(18)

Supreme Court, U.S.
FILED
DEC 17 1998
OFFICE OF THE CLERK

No. 98-377

In the Supreme Court of the United States

OCTOBER TERM, 1998

BRUCE A. LEHMAN,
COMMISSIONER OF PATENTS
AND TRADEMARKS, PETITIONER

v.

MARY E. ZURKO, ET AL.

ON WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

JOINT APPENDIX

SETH P. WAXMAN
Solicitor General
Department of Justice
Washington, D.C. 20530-0001
(202) 514-2217

Counsel of Record for
Petitioner

ERNEST GELHORN
Suite 100
2907 Normanstone Lane, NW
Washington, D.C. 20008-2725
(202) 319-7104

Counsel of Record for
Respondents

PETITION FOR WRIT OF CERTIORARI FILED: AUGUST 31, 1998
CERTIORARI GRANTED: NOVEMBER 2, 1998

110 pp

TABLE OF CONTENTS

	Page
Relevant Docket Entries:	
United States Court of Appeal for the Federal Circuit	1
Patent and Trademark Office	3
Record Materials:	
Specification, as filed with patent application No. 07/479,666 (February 13, 1990)	5
Information Disclosure Statement (#2) (May 1, 1991)	67
Preliminary Amendment, included as part of Preliminary Amendment and Supplemental Information Disclosure Statement (May 1, 1991)	72
Second Amendment, included as part of Response to Office Action of March 18, 1992 (June 19, 1992)	79
Brief in Opposition to Petition for Rehearing and Suggestion of Rehearing En Banc (June 20, 1997)	84

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

Docket No. No. 96-1258

IN RE MARY E. ZURKO, ET AL.

RELEVANT DOCKET ENTRIES*

DATE	PROCEEDINGS
2/1/96	Notice of appeal filed.
	* * * * *
11/13/96	Oral argument before panel.
	* * * * *
4/15/97	Opinion and judgment: Reversed.
	* * * * *
7/3/97	Suggestion for rehearing en banc granted.
	* * * * *
12/2/97	Oral argument before court in banc.

---

* Entries edited for clarity.

5/4/98	Opinion and judgment: Reversed.
	* * * * *
9/10/98	Received notice that petition for writ of certiorari filed with Supreme Court, 8/31/98.

UNITED STATES DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE

IN RE PATENT APPLICATION NO. 07/479,666

RELEVANT DOCKET ENTRIES*

DATE	PROCEEDINGS
2/13/90	Application filed.
	* * * * *
5/2/91	Preliminary Amendment and Supplemental Information Disclosure Statement received.
	* * * * *
3/18/92	Initial rejection (revised).
	* * * * *
6/23/92	Response to Office Action of March 18, 1992 (including second amendment) received.
9/8/92	Final rejection.

---

* Entries edited for clarity.

12/1/92	Notice of appeal.
	* * * * *
7/31/95	Decision by Board of Patent Appeals and Interferences: Affirmed.
	* * * * *
12/1/95	Reconsideration denied.
2/1/96	Notice of appeal to United States Court of Appeals for the Federal Circuit.
	* * * * *

PD90-0141
DIGU:012/TOE

METHOD FOR EXECUTING TRUSTED-PATH COMMANDS

BY

Mary Ellen Zurko
Thomas A. Casey, Jr.
Morrie Gasser
Judith S. Hall
Clifford E. Kahn
Andrew H. Mason
Paul Douglas Sawyer
Leslie Richard Kendall
Steven B. Lipner

TABLE OF CONTENTS

		PAGE
1.	BACKGROUND OF THE INVENTION	[7]
1.1	Introduction	[7]
1.2	Background	[7]
1.3	Security Criteria; Reference Validation Systems	[7]
1.4	Assurance Objectives and "Trusted" Systems	[11]
1.5	Parsers as Increasers of System Complexity	[13]
2.	SUMMARY OF THE INVENTION	[15]
3.	BRIEF DESCRIPTION OF THE DRAWINGS	[20]
4.	DESCRIPTION OF SPECIFIC EMBODIMENTS	[21]
4.1	Overview of Software Architecture	[21]
4.2	User Terminals and the Secure-Attention Key	[24]
4.3	The Trusted Path and Direct Commands	[26]
4.4	Login Processing by the TCB	[28]
4.5	Connection to Untrusted Process	[29]
4.6	The Process Identifier	[30]
4.7	The Command Conveyor	[34]
4.8	Distribution of Processing of Trusted Commands	[35]
4.9	Display of Command to User	[40]
4.10	The SSVR as an Audit Recorder	[48]
4.11	Confirmation by Others than the Submitting User	[49]
CLAIMS		[52]
ABSTRACT		[59]

1. BACKGROUND OF THE INVENTION

1.1 Introduction

This invention relates to a method for improving security in a computer system. More specifically, this invention concerns a method for implementing trusted commands through both trusted and untrusted code.

1.2. Background

The proliferation of computers has resulted in the need for reliable computer security systems. The need to protect certain information is great in areas dealing with national security, for example. Security measures are required in other fields, including areas which utilize financial, medical, and personal information. Many computerized information systems therefore include internal security measures which provide the users with different levels of access to the computerized information, and which attempt to provide at least a degree of protection of the information from undesirable access.

1.3. Security Criteria: Reference Validation Systems

In response to the need for secure computer systems, the Department of Defense has issued a publication titled Department of Defense Trusted Computer System Evaluation Criteria (reference No. DOD 5200.28-STD). This publication is sometimes referred to as the "Orange Book," and is available from the Department of Defense. The Orange Book describes

system security objectives and evaluation criteria for secure computer systems.

A "secure" computer system generally includes some type of "reference validation" system. These reference validation systems (known as reference monitors) are responsible for enforcing the security rules (security policy) for a given computer system.

Reference monitors mediate all access to "objects" by "subjects". Objects are passive entities that contain or receive information. Access to an object implies access to the information that it contains. Subjects are active entities, generally in the form of a person, process or device that cause information to flow among objects or change the system state. A subject may ordinarily reference its own, subject-internal information without the involvement of the reference monitor. Reference monitors controlling such access to objects by subjects are known in the art, and may utilize a security kernel approach.

Proper implementation of a reference monitor calls for adherence to three principles:

(1) completeness, in that all access by subjects to objects or other subjects must involve the monitor;

(2) isolation, in that the monitor must be protected from tampering; and

(3) verifiability, in that correspondence must be shown between the security policy and the actual implementation of the monitor.

As discussed, every reference to information or change of authorization should go through the reference monitor. Thus, all commands issued by a user or other subject are monitored by the reference monitor. This approach is particularly useful in multiuser computer environments.

The totality of the protection mechanisms within a computer system—including hardware, software, and firmware, the combination of which is responsible for enforcing a security policy—is commonly known as a "trusted computing base" (TCB). If the trusted software is designed to be as simple as possible, for the sake of verifying the reference monitor, then the trusted software is known as a "security kernel".

Generally, TCBs attempt to meet the control objectives set out in the Orange Book. Compliance with these objectives inspires user confidence, and increases the overall desirability of a computer system. These objectives deal with:

(1) security policy;

(2) accountability; and

(3) assurance.

The security policy objective entails enforcement by the TCB of the desired security rules. These security rules are designed to limit the access to and dissemination of information in a precisely defined manner.

Security policies may include provisions for the enforcement of both mandatory and discretionary access

control rules. Mandatory access control rules control access based directly on comparisons of the user's security level and the sensitivity level of the information being sought. Discretionary access rules control and limit access to identified individuals who have been determined to have a need-to-know.

These access control rules call for associating with each user identification code a statement indicating the user's access rights. This statement often includes information representing the user's security level (for mandatory control purposes), and membership in groups (for discretionary control purposes).

The accountability objective calls for providing each user with an individual user identification code (often called a "user name") and for the TCB to be able to recognize the code and ensure that it is being used by its proper user. This may be done by checking the user's password. This ensuring the user's identity is known as "authentication."

In addition, the accountability requirement calls for the existence of auditing capabilities. Such capabilities allow for the auditing of actions which can cause access to, generation of, or release of classified or sensitive information.

1.4 Assurance Objectives and "Trusted" Systems

The assurance objective is especially important in the present context. That objective is concerned with taking steps to ensure that the security policy is correctly implemented and that the TCB accurately mediates and enforces the intent of that policy. Steps may be taken to insure that each portion of the TCB is assured. To accomplish this objective, two types of assurance are needed.

The first type of assurance is life-cycle assurance. This type of assurance refers to steps taken to ensure that the computer system is designed, developed, and maintained using formalized and rigorous control standards.

The second type of assurance is operational assurance. Operational assurance deals with the features and system architecture used to ensure that the security policy is uncircumventably enforced during system operation. All of the software (sometimes referred to informally in the art as "code") in the TCB is generally analyzed to determine the assurance level of the system.

As the amount of code in the TCB increases, it becomes more difficult to ensure that the TCB accurately enforces the security policy. Because of this, it is desirable to minimize the amount of trusted code, and thus the complexity of the TCB.

A TCB is usually operated with a substantial amount of software, such as text editors and other applications, operating within the security policy of the TCB.

Generally, this untrusted software asks the TCB for access to objects when the user or the untrusted software requires them. Thus, the majority of user's requests to the TCB, and the majority of the information that a user obtains from the TCB, are handled through the agency of untrusted software.

This untrusted software, however, is by nature in danger of compromise and vulnerable to bugs. For some types of requests and displays, malicious or malfunctioning untrusted software could compromise the enforcement of the security policy. Generally, TCBs cannot distinguish between requests faithfully made by the untrusted software on command from a user and either requests made by the untrusted software at its own initiative or requests that misrepresent the user's actual command. For example, if commands issued by an authorized user to change certain users' security levels were made through the agency of untrusted software, it would be possible for malicious or malfunctioning untrusted software to inappropriately raise the security level of a user. Such inappropriate raising of a security level could result in the disclosure of sensitive information.

Furthermore, TCBs generally cannot ensure that displays made by untrusted software are faithful. This poses problems in that if displays of audit records were made through the use of untrusted software it would be possible for malicious untrusted software to misrepresent these audit record to hide suspicious activities.

To overcome these problems, prior-art systems have developed a concept known as a "trusted path." A trusted path is a mechanism by which the user can

communicate directly with the TCB. A trusted path may only be activated by the user or the TCB and cannot be imitated by untrusted code. Such a trusted path is a positive TCB-to-user connection that bypasses all untrusted software. The Orange Book requires a trusted path for all systems to have a security rank of B2 or above.

A "trusted command" (also known as a trusted-path command) is a command which requires a trusted path between the user and the TCB for execution. The specific security policy of a computer system will determine which commands are defined as trusted commands. For example, commands relating to changes of the security levels of users would be trusted commands.

1.5 Parsers as Increasers of System Complexity

Because of perceived performance problems, prior-art computer systems often included code to implement certain functions in the TCB. One such function comprises a portion of code known as the "parser."

The parser performs the basic interpretation of user commands by translating a human-readable representation of a user command (e.g., a command typed by a user at a keyboard) into a machine-readable representation known as the binary representation, or the parsed command. A user command may consist of a group of words, typed by the user, specifying some action to be taken. The user command may also specify one or more variations or options of that action.

For the convenience of the user, most parsers allow these option words to be typed in any order without changing the meaning of the user command. Also, most parsers allow the user to type the words in either full or abbreviated form. In some instances, several possible abbreviations exist. Further, in most parsers, the user may vary the spacing between the words without changing the meaning of the command. Some parsers also check user commands for proper syntax, and report errors to the user prior to execution of the command. Other parsers prompt the user for missing information, avoiding the need to completely retype an incomplete command.

Thus, several different inputted human-readable representations may be associated with each unique parsed command. It is the job of the parser to translate accurately these inputted representations of user commands (which may vary with each user) into specific parsed command representations. Due to the complex nature of parsing, large amounts of computer code are generally associated with this activity.

Prior-art trusted software systems have included the parser for trusted commands, and thus the associated code, within the TCB. In these systems every trusted command was parsed and executed exclusively by trusted code. The inclusion of the parsing code in the TCB was regarded as necessary for proper system operation.

As discussed above, however, the ease and degree of assuring a system is roughly inversely proportional to the complexity of the system's TCB. Thus, the inclusion of the parser in the TCB results in more

complex assurance analysis and a corresponding decrease in system confidence. Prior art devices (since they needed to be small and simple to support assurance) have thereby inherently constrained the user-friendliness of computer systems.

2. SUMMARY OF THE INVENTION

The present invention provides a number of advantages over the prior art. By reducing the complexity of the TCB, the amount of trusted code that must be verified or assured is reduced. In addition, the general usability of the system is increased, because the improved method of processing trusted commands allows for the implementation of added computing functions at little or no additional cost in TCB assurance.

The present invention includes an improved method for the execution of trusted commands. In this improved method, the required steps for processing a trusted command are split between trusted and untrusted code. In this manner the amount of trusted code in the TCB may be significantly reduced.

As discussed above, the parsing of a command can be comparatively difficult, and often requires large amounts of complex code. However, it is relatively simple to translate the parsed representation of a command into a standard human-readable representation for display to the user. This is in part because a single, standard human-readable display representation may be associated with each parsed representation. The present invention takes advantage of this fact by

parsing the trusted commands in untrusted code prior to execution in the TCB.

In general, the present invention includes a computer system which includes a TCB operating as a reference monitor. Included in the TCB is the security-relevant code necessary for the actual execution of a trusted command.

Associated with the TCB are one or more untrusted subjects (e.g. processes). Each such subject has particular access rights, which may include a security level. These untrusted subjects communicate only with the involvement of the TCB. The TCB allows communication only as permitted by the system's security policy. Each untrusted subject is substantially independent.

In each of these untrusted subjects, untrusted software is generally run. This untrusted software may include an untrusted operating system (UOS), and generally includes applications such as test editors. In the present invention, the untrusted code includes the non-security relevant functions necessary for the execution of a trusted command.

Attached to the computer system are several user terminals. These user terminals allow the user to exchange information with the computer system. All terminal activity is seen first by the TCB. A secure-attention key is dedicated for use on each terminal attached to the computer system. The TCB is designed to recognize and trap this key signal. Activation of this key by the user establishes a trusted path between the

user terminal and the TCB. Alternate methods for establishing a trusted part may be utilized.

The TCB is generally provided with a limited number of "direct" user commands (direct commands). Because of this, the majority of the user's activities will take place in the untrusted operating system.

While operating in the UOS, the user may attempt to execute a trusted command. When one of these commands is issued by the user, the untrusted code in the UOS checks the syntax of the command and may prompt for missing parameters; then the untrusted parser converts the command into a parsed representation that will be referred to here for convenience as the "binary representation." In some instances, command procedures and application-issued trusted commands will also be parsed and translated into a binary representation.

This binary representation is then passed from the UOS to the TCB. The TCB initially associates the binary representation with a physical terminal, and copies the binary representation into a memory buffer assigned to that terminal.

Once this has been completed, the untrusted code in the UOS may prompt the user to activate the secure-attention key. If this key is activated, a trusted path will be established between the user terminal and the TCB. Alternate methods may be employed to initiate this trusted path. A randomly generated process ID, which may be generated at login, may be utilized to prevent untrusted code from simulating a trusted path.

Subsequent activity by the TCB depends on the type of command that is represented by the representation. If the command is one that will not modify any information, i.e., a request to view information, the TCB verifies that the user has the necessary access rights, and attempts to execute the binary representation. A human-readable standard representation of the binary command received by the TCB is displayed along with the information.

In most instances, if the command requested by the user requests modification of information, the TCB displays for confirmation a standard human-readable form of the command, along with an indication of what the result of command execution will be. The user is then requested to confirm (by typing a response via the trusted path) that the displayed command was received correctly and should be executed.

This confirmation allows the user to ensure that untrusted code has not performed an unauthorized modification of the user's original command. Also, confirmation allows for the detection by the user of any errors that may have occurred during the parsing process. In situations where the binary representation represents a command procedure or an application-issued trusted command, the confirmation ensures that the command proposed by the untrusted software is acceptable to the user.

If the command is confirmed by the user, the TCB checks to ensure that the user has the necessary access rights to execute the command and, if so, attempts to execute the command.

A distinct advantage of the present invention arises from the fact that the execution of a trusted command is split between trusted code and untrusted code. The actual parsing of the command occurs in untrusted code. The trusted code included in the TCB checks the parsed command of correctness and displays it to the user in human readable form. This checking and display of a parsed command requires substantially less code than is required to parse a command.

Thus, since the command parser is included in the untrusted code, the amount of trusted code in the TCB is minimized. This lowers the difficulty of trusted system assurance, and increases the amount of confidence that may be placed in a system.

It is a further advantage of this invention that the user interface is separate from the underlying functions performed by the TCB. This allows for implementation of alternate UOS-to-user interfaces without major modification of the TCB. Further, the amount of time and effort to develop such a command interface is reduced. A still further advantage is that more useability features may be provided without substantially increasing the complexity of the TCB. Examples of such useability features are command line editing and recall.

3. BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates the various levels within a computer system;

Figure 2 illustrates various layers within the trusted computing base;

Figure 3 illustrates a trusted path;

Figures 4A and 4B are flow diagrams representing a method for processing trusted commands. Differences in box representation in these drawings is explained below.

Figure 5 illustrates one method of implementing the invention;

Figure 6 is a flow diagram representing a method for verifying a trusted path;

Figure 7 is a state transition diagram of a further embodiment utilizing various command states.

4. DESCRIPTION OF SPECIFIC EMBODIMENTS

A specific embodiment of the present invention is described with respect to a computer which supports various user processes and may operate in a multiuser environment. As an illustration, an implementation in connection with a VAX model computer, available from Digital Equipment Corporation (DEC), is briefly described. It should be understood that the present invention may be readily applied to various types of computer systems including PC networks, stand-alone multiuser computer systems, or single user computer systems, to name a few examples.

4.1 Overview of Software Architecture

FIG.1 illustrates a preferred computing environment which is generally designated by the reference numeral 2. A trusted computing base (TCB) 10 resides at the center of the computer system. The TCB 10 acts as a reference monitor in that access to information is mediated by the TCB 10.

In a general embodiment of the invention, the TCB 10 may be a secure kernel residing under the general operating system of a computer system. In one embodiment, the TCB 10 comprises trusted code residing underneath a VMS or Ultrix system.

The TCB 10 enforces a security policy which identifies permissible modes of access between users or computer processes (subjects) and files, memory segments or other objects. Access to stored information is mediated by the TCB 10, so that the TCB 10 may

determine whether a particular user or untrusted subject can access particular information. In order for the TCB 10 to fulfill its protective function, it must take into account the three requirements discussed above: (1) completeness, (2) isolation, and (3) verifiability.

A more complete understanding of the TCB 10 may be obtained by reference to FIG.2. As is illustrated, the TCB 10 may comprise several distinct layers. Each layer of the TCB 10 comprises trusted code. This trusted status is indicated by the security boundary 11. In one embodiment, layers of particular interest are layers to be referred to as: secure server (SSVR) 12, Command Conveyor (CC) 14, and VTerm (VT) 16.

The TCB 10 includes the trusted code necessary to ensure compliance with the Orange Book requirements. In alternate embodiments, the trusted code may ensure compliance with different systems of security requirements. The TCB 10 also includes the code which actually executes the requested trusted command, as discussed below.

Each untrusted subject is generally supported by an untrusted process. This process may spend time in one of two "modes." The first mode, in which the process usually runs, is known as a "user mode." This mode is a restricted hardware state such as the VAX user mode that is known in the art. This process may also spend time executing trusted code in an all-powerful hardware state. This state is represented by the VAX "kernel mode", which is known in the art. This state may be triggered by a request by the untrusted subject of the security kernel. Such a request may be made in a conventional manner and is commonly called a "kernel

call." This mode may also be triggered by asynchronous events (e.g. the completion of an Input/Output request previously made by the untrusted subject).

In one embodiment, the TCB 10 also has one process associated with each terminal that communicates with the user over the trusted path. This process is known in the art as an "execution thread." In this embodiment, each execution thread shares its address space with other TCB execution threads. An advantage of such sharing is economy of machine resources. Full-blown processes, each with its own address space, would require far more machine resources than if such address space sharing is employed.

It is the CC 14 which provides the interprocess communication for passing parsed commands between the processes that support the untrusted subjects, and the processes within the TCB 10 that support the trusted path.

The untrusted code may submit to the TCB 10 the parsed representation of the trusted command by making a kernel call. The kernel call causes the process supporting the untrusted subject to switch to the all-powerful hardware state and to begin to execute trusted code designed to handle kernel calls. This trusted code recognizes the nature of the kernel call and calls the CC 14. The CC 14 then obtains and copies the binary representation of the trusted command into protected memory and puts it in a global data structure that supports the user process associated with the terminal in question.

The VT 16 layer includes the trusted code which monitors all terminal activity and checks for activation of the secure-attention key. In one particular embodiment, the VT 16 layer also includes code which tracks which physical terminal is associated with a particular user or untrusted subject.

The specific manner of implementation of the above described activities will be apparent to one skilled in the art having the benefit of this disclosure.

Referring to FIGS 1 and 3, "around" the TCB 10 is a general untrusted operating system (UOS) 20. The UOS 20 generally comprises untrusted code and may include a traditional operating system such as Ultrix or VMS. Alternate constructions are envisioned where the UOS 20 includes any untrusted code. It will be appreciated that the system may include untrusted code that is separate from TCB 10 and UOS 20.

Most user activity will occur in the UOS 20. In one embodiment, a specific UOS 20 may run in each untrusted subject.

4.2 User Terminals and the Secure-Attention Key

Associated with the computer system are one or more user terminals 30. These terminals 30 may be conventional in construction, in that they allow the user to exchange information with the computer system. Either "intelligent" or "dumb" terminals may be used. Examples of such dumb terminals are the well-known VT series 200 and 300 terminals. These terminals are available from Digital Equipment Corporation or its authorized dealers.

A particularly useful terminal includes control programming that is designed with security features in mind, such as is described in a co-pending U.S. patent application entitled "Method for Securing Terminal and Terminal Apparatus for Use with the Method," Serial No. 456,672, filed by Wei-Ming Hu, Clifford E. Kahn, Andrew H. Mason, and Peter A. Sichel on December 26, 1989 and to be commonly assigned with this application.

A keyboard is attached to each user terminal. Each keyboard includes a key employed as a secure-attention key. This key is utilized to establish a trusted path between a user terminal and the TCB 10. In one embodiment, the secure-attention key is an "out-of-band key" for the UOS 20. It is desirable that the signal sent by the terminal upon activation of the secure-attention key is generally not utilized by the applications running in the UOS 20. An out-of-band key is convenient since it does not restrict the traditional number of user keys and the signal is sent immediately, even if other I/O is pending. For example, the secure-attention key may be the BREAK key on the terminal keyboard.

In many terminals (such as the VT 2XX and VT 3XX families) locking of the keyboard prevents the transmission of several key signals. In these terminals, however, the BREAK key signal may be transmitted, even from a locked keyboard. Thus it is generally desirable to employ the BREAK key as the secure-attention key. Also, the signal generated by activation of the BREAK key comes in ahead of other characters in the keyboard buffer. A further advantage of utilizing the BREAK key is that software running in the UOS 20 cannot get the terminal to send that character.

4.3 The Trusted Path and Direct Commands

The TCB 10 generally operates as a reference monitor.

Upon initial login or at other times when the user is not connected to a process in the UOS 20 (e.g., an application program), the TCB 10 responds to a comparatively limited series of "direct commands," which may include a CONNECT command to connect to an untrusted subject. When the user is connected to such an untrusted subject, the TCB 10 checks for the secure-attention signal; if a signal other than the secure-attention signal is received, the TCB 10 passes the signal to the connected untrusted subject.

As discussed above, in one specific embodiment the trusted code in the VT layer 16 monitors all terminal activity. When the code in the VT 16 notices that the BREAK key has been depressed, the SSVR is signaled in a conventional manner (such as by advancing an event count or activating a flag). Code in the SSVR 12 recognizes this signaling and establishes a trusted path between the user terminal and itself. In this manner, the secure-attention key signal is trapped by the TCB 10.

In certain security environments, general security requirements mandate that the TCB 10 makes displays when a trusted path is established and exited by the user. In one embodiment, when a trusted path is established, the TCB 10 sends a reset command sequence to the physical user terminal 30 to assure that it is in a known state. This is done in part to ensure that the information displayed to the user by the TCB 10 is in

the intended form, (e.g., correct font). A particularly useful sequence of commands to implement the reset command sequence is described in the aforementioned copending U.S. patent application by Hu et at.

When the secure-attention key is activated, other than in connection with the issuance of a trusted command, the secure server 12 displays a particular banner and prompt (for example: "SSVR" together with the user's process identifier, discussed below), to indicate that the user is conversing interactively with trusted code. If the secure-attention key was activated in connection with the issuance of a trusted command, the secure server 12 displays to the user some indication of what is to be done, as discussed below.

When command execution is complete, or when the user establishes or reactivates a session with the UOS 20, the TCB 10 issues a message to inform the user that he/she is breaking the trusted path. When a session with the TCB is complete, the memory in the physical terminal may be cleared by the TCB. This is done to protect trusted information from both untrusted code and unclassified persons who may later use the terminal. A particularly advantageous method of clearing the terminal memory is described in the aforementioned co-pending patent application.

The number of direct user commands available from the TCB 10 is preferably limited to reduce the complexity of the TCB 10. Generally, a command is direct if:

a. one needs the command when one is not connected to an untrusted subject;

b. end users depend on the command's correct behavior for security (e.g., LOGOUT); or

c. the command must be issued when no user process in the UOS 20 is running.

In most cases, the user will converse with the TCB 10 through the SSVR 12. In one embodiment the user conversing with the SSVR 12 may be restricted to commands which allow him/her to establish (connect) or end (disconnect) a computing session in the general UOS 20. Further commands may be added, such as allowing the user to view the existing UOS 20 session or change his or her password, but it is generally desirable to limit the number of available commands in the TCB 10. It is further desirable to ensure simplicity in the syntax of these direct commands.

The simple syntax of the direct commands, and the comparatively small amounts of code necessary to interpret these commands, do not substantially increase the complexity of the TCB 10.

4.4 Login Processing by the TCB

To initiate a computing session in the illustrative UOS 20, the user may first login to the computing environment through the SSVR 12. To initiate a login, the user activates the secure-attention key (or performs some other action to gain the attention of the SSVR 12). As discussed above, this establishes a trusted path between the user terminal and the TCB 10. Alternate embodiments are envisioned wherein other methods are employed to establish a trusted path.

Once a trusted path has been established, the SSVR 12 prompts the user for his/her specific user name and password. The various layers in the TCB 10 perform well-known security checks to ensure that the user has the necessary access rights to login at the terminal. If the user possesses the required access rights, a computing session is established in the SSVR 12.

At this time, the user's security information is stored by the TCB 10. The TCB 10 also stores in memory a unique identifier of the user process. This information is maintained in memory by the TCB 10 as long as the user has an active session.

4.5 Connection to Untrusted Process

Because the number of director user commands in the TCB 10 is preferably limited, the user will generally choose to establish a computing session with an untrusted subject. Such an untrusted subject may include application programs residing in the UOS 20.

In one specific embodiment, when the user establishes a computing session in the UOS 20, the user's physical terminal 30 is associated with a virtual terminal of the UOS. The user converses with the untrusted subject through a computing session within the UOS 20. When such a connection with an untrusted subject is made, the VT layer 16 of the TCB 10 stores in memory which physical terminal 30 is associated with which virtual terminal in an untrusted subject. In this manner, trusted commands received from an untrusted subject may be attributed to the actual user at the physical terminal 30. Once the user has established a

computing session within the UOS 20, normal computing activities may be carried out.

A more detailed understanding of these activities may be obtained through reference to FIG.4A and 4B. Activities indicated by heavily bordered blocks are performed by the TCB 10, or trusted code. Activities represented by lightly bordered blocks are performed by untrusted code.

In FIG.4A, the user terminal is represented as having established a computing session 100 within the general UOS 20. During the session, the user may attempt to execute various commands. Utilities within the UOS 20 determine whether the issued commands are trusted commands 50. If the command is not identified by the UOS 20 as a trusted command, the UOS 20 attempts to perform the command at process 60. In these cases the operation and existence of the TCB 10 may have little effect on the command's behavior as seen by the user.

4.6 The Process Identifier

In one embodiment, a process identifier (process ID) is associated with each user process at the time of login. This process ID is a pseudo-randomly generated alphanumeric tag which is associated with the user throughout his/her computing session.

The process ID is useful if the untrusted code can temporarily disable or delay the effect of activation of the secure-attention key. For example, untrusted code can send a reset command to some terminals. While these terminals are resetting, activation of the secure-

attention key has no effect. Such a technique may allow untrusted code to trick the user into thinking a trusted path had been established when it had not been. The process ID inhibits such trickery by allowing the user to distinguish between actual and emulated trusted paths.

When the user first logs in, he/she will not yet know the process ID and cannot detect a false trusted path. However, to simulate a login dialogue, untrusted code would have to guess when the user activated the secure-attention key, or somehow detect the activation of the key even though the TCB could not detect it. Guessing the activation time is impracticable, because users start login dialogues at their own convenience and give no advance signal to the TCB prior to activation of the secure-attention key. Detecting the secure-attention key even though the TCB could not detect it would be possible only by causing the key to yield a signal different than the TCB expects. Methods to prevent such an event are described in the co-pending application by Hu et al., referenced above.

With some terminals, the untrusted code may be able to program the terminal so that the secure-attention key generates a signal that is recognizable to the untrusted code, but not to the TCB. In these cases, generation of the process ID at the time of login would be unsatisfactory. Instead, the process ID may be generated prior to user login. For example, in these embodiments, the process ID may be generated when the user is registered with the TCB 10 and stored in protected memory. In this embodiment the user may be given a way of changing his/her process ID, just as the user can generally change their password.

In one particular embodiment, when the user logs in the SSVR 12 psuedo-randomly generates a process ID and assigns this ID to the specific user at the physical terminal. Methods of generating psuedo-random identifiers are generally known in the art. In one embodiment, the random number initial seed is based on a hardware (as opposed to a software) event. The algorithm for generating successive random numbers should be computationally difficult to invert.

This process ID is maintained in the TCB and is never provided to untrusted code. Thus, untrusted code in the general UOS 20 is prevented from discovering the specific process ID assigned to the user.

The user is responsible for noting and retaining his/her assigned process ID. To promote user retention the process ID may be generated as a pronounceable group of characters. Methods for such generating pronounceable passwords are described in "A Random Word Generator", MITRE Technical Report (MTR-3006), May, 1975, by Morrie Gasser.

The process ID is displayed to the user when a trusted path is established between the user terminal and the SSVR 12. Displaying the ID serves a useful security function as it inhibits untrusted code from "spoofing" the user. Absent the process ID it may be possible for an untrusted subject to emulate the SSVR 12. An untrusted subject, emulating the SSVR 12, may be able to trick the user into divulging security-related information or otherwise compromising system security.

The process ID, known only to the user and the SSVR 12, inhibits this type of "spoofing." Before the user responds to an apparent SSVR 12 request, he/she will first verify that the proper process ID is displayed. Since the process ID is known by only the user and the SSVR 12, it is extremely difficult for an untrusted subject, ignorant of the process ID, to fool the user into believing that he/she is conversing with the SSVR 12.

FIG.6 illustrates a flow diagram of this process. When the user initially logs into the computer system through the SSVR 12, a randomly generated process ID is assigned to the user. This activity is represented by process 310. In one embodiment, this process ID comprises alphabetic characters and is pronounceable. The pronounceable nature promotes user retention of the ID. The process ID is then stored in trusted memory by the SSVR 12 and displayed to the user by the SSVR 12 at process 320.

During the course of the computing session, the TCB determines if the user is operating through a trusted path. This activity occurs at process 330. If the TCB determines that a trusted path is not established, the process ID will not be displayed to the user. This is noted at process 350.

The process ID is maintained in trusted code and memory, and is inaccessible to untrusted subjects. Each time a trusted path is established between the SSVR 12 and the user, the SSVR 12 displays the user's process ID at process 340. By observing the displayed ID, the user is assured that an actual trusted path has been established.

4.7 The Command Conveyor

As discussed above, the command conveyor (CC) 14 is a layer in the TCB 10 which comprises part of the trusted code. The CC 14 is utilized to ensure proper communication of trusted commands between the SSVR 12 and the process supporting the untrusted subject. As discussed, such an untrusted subject may include a UOS 20.

Associated with each physical terminal 30 is a command buffer. In some embodiments, a status code may be associated with each terminal. In a still other embodiments, one of five command states is also associated with each physical terminal. The command buffer is used to store the binary representation of the trusted command. The status code may be employed to provide the resulting execution status to the untrusted subject. The command states generally comprise five possibilities:

(a) no command,

(b) command submitted,

(c) command in progress,

(d) ignore finish, and

(e) command done.

These states are used to indicate the various states that may encountered in the execution of a trusted command. FIG. 7 is state transition of the command

states. A detailed explanation of these states and the statuses may be obtained from the following discussion.

4.8 Distribution of Processing of Trusted Commands

As discussed above, the TCB 10 "recognizes" two categories of commands. For the purposes of this specification "commands" include user instructions received through the use of command line interfaces, menu interfaces, direct manipulation interfaces or the like.

The first category includes "direct commands" issued while the user is operating in the SSVR 12 layer of the TCB 10, i.e., is interactively conversing with the TCB 10. As discussed above, it is desirable to limit the number of direct commands available to the user. The execution and processing of these direct commands is carried out using traditional methods known in the art.

The second category includes "trusted commands." These commands are issued by a user operating in a non-trusted computing environment, such as in the general UOS 20. In one embodiment these trusted commands are issued by a user operating in an untrusted virtual machine. FIGS.4A and 4B illustrate the activity necessary to execute such a command.

If the user attempts to execute a trusted command, the parser residing in the general UOS 20 parses the trusted command string and converts it into a binary representation at process 70. In parsing the command, the untrusted code in the UOS 20 checks syntax and semantics, and prompts for missing parameters. Pars-

ing strategies are well known in the art and will not be discussed herein.

From a security perspective, it is not required that the code in the UOS 20 perform an accurate translation of the user's commands into a parsed representation (binary representation). The confirmation mechanism, described below, is designed to catch such errors whether benign or malevolent.

Following parsing of the command, a process supporting the UOS 20 issues a call to the TCB, submitting to the TCB the identification of the terminai from which such command was issued, at process 80. In one embodiment, the virtual terminal from which the command was issued is submitted to the TCB. Such a call is processed by the CC 14 in the TCB 10. The CC 14 then determines which trusted subject submitted the command and copies the parsed representation into protected memory at process 90. In one embodiment this determination is made by asking V Term to identify the physical terminal associated with the UOS's virtual terminal.

Such a submission of the binary representation will be successful only if the TCB 10 determines that the submitting terminal is connected to an active process.

Normally, in embodiments utilizing the command states, the command state of the submitting physical terminal is no command, which indicates that there is no trusted command submitted by that physical terminal or in progress. This state is represented by state 400 in FIG. 7. When the state is no command, the CC 14 saves the binary representation in the command

buffer, and sets the physical terminal's state to command submitted, represented by transition state 410. Otherwise, the submission fails.

When the untrusted subject running in untrusted code receives indication from the TCB 10 that the command has been submitted, it requests the user to activate the secure-attention key. In one embodiment, the untrusted code in the UOS 20 will request that the user activate the BREAK key.

If the user activates a cancellation key, for example Ctrl-Y or Ctrl-C, instead of activating the secure-attention key, code in the general UOS 20 may cancel the trusted command. This can be accomplished by the untrusted code issuing a kernel call (i.e., a request by the untrusted subject of the security kernel) and informing the CC 14 finds that the command is to be aborted. If the CC 14 finds that the command has not been sent to the SSVR 12, the command is canceled. Similarly, if the user process in the UOS 20 is killed, the process rundown code may cancel the trusted command request. However, if the CC 14 finds that the command has already been sent it proceeds; because the user is already executing the command.

In embodiments employing command states, cancellation of the trusted command request results in the command state being changed from command submitted to no command. This process is indicated in FIG. 7.

If the user attempts to establish a trusted path at the same time the trusted command is sent to the TCB 10, then the TCB 10 must arbitrate; either of these re-

quests may be processed first. If the secure-attention signal is processed first, the TCB 10 may defer the command submission, and a connection with the SSVR 12 is established. If the submission is processed first, then the user sees the confirmation screen as is described below.

If, when prompted, the user activates the secure-attention key, a trusted path is established between the user's physical terminal 30 and the SSVR 12 in the TCB 10. This configuration is illustrated by the dark line in FIG.3. When the VT layer 16 detects the secure-attention (BREAK) key, it notifies the SSVR 12 which then establishes a trusted path in the trusted path-supporting process associated with the user's physical terminal 30.

Once the trusted path is established, the TCB 10 maintains exclusive control over the physical terminal 30. In one embodiment, untrusted code in the UOS 20 waits to be notified that the command has been completed. Thus, the user essentially bypasses the untrusted code, and converses only with trusted code. Thus, if the user's process in the general UOS 20 is killed, or the user's process terminates, or the general UOS 20 crashes, the trusted command may still be executed. The user will not be informed of any of these activities until the trusted command has been completed or abandoned.

Following establishment of the trusted path, the security requirements of the system require interaction between the user and the SSVR 12. This is necessary to execute commands which require the establishment of a trusted path. The Orange Book criteria, and/or the

system's security goals, will determine which commands require this interaction as is generally discussed in Section 4.9 below.

The establishment of a trusted path in these instances is useful for several reasons. First, it allows the secure server to identify the specific user issuing the command. Second, the trusted path allows the user to review and approve (or reject) the requested action in a trusted fashion. This reduces the possibility of malicious untrusted software altering a command request, and allows for the detection of parsing errors.

FIG.4B illustrates the activity occurring after control has passed to the TCB 10. As discussed above, the TCB 10 first establishes a trusted path at process 120. The TCB then checks to see if a trusted command has been submitted by the user. Untrusted code may then wait for the command to finish.

Once a trusted path has been established, the SSVR 12 will call the CC 14 to return the binary representation at process 130. In one embodiment, the CC 14 will first determine the command state of the physical user terminal 30. In a still further embodiment if the command state is *command submitted* 410 (FIG. 7), the CC 14 will return to the SSVR 12 the binary representation, and change the command state to *command in progress* 420. If the state is *no command* 400 or *command done* 430, the call will fail, and the SSVR 12 will display the SSVR prompt (SSVR>) to the user. An initial command state of *Command in progress* 420 or *ignore finish* 440 implies a bug in the SSVR 12 and may result in the CC 14 crashing the TCB 10.

In these embodiments, if the command state was command submitted 410, then the CC 14 will return to the SSVR 12 the binary representation of the requested command.

4.9 Display of Command to User

Once the parsed command has been returned to the SSVR 12, the SSVR 12 will determine if the command requires confirmation at process 140 (FIG. 4B). Confirmation should generally be required for commands which request modification of information. Commands that do not alter the information, such as the viewing of information, should not generally require confirmation. For purposes of confirmation, redirecting the display into a file should be considered as viewing information.

The decision of whether a given command should require activation of the secure-attention key to execute depends on the basic security policy of the system. In one embodiment, secure-attention trusted commands are a mechanism for use by system managers, security managers, operators, and holders of special privileges.

Generally, a command which is not a direct command, is a secure-attention command if:

a. it modifies a security kernel database in a security-relevant way; or

b. it displays information on which a systems manager, security manager, or operator will base a security decision.

If the command is one that does not require confirmation, the TCB 10 enforces the system's security policy and determine if the user has the necessary access rights to execute the command at process 145. If this is the case the TCB 10 attempts to execute the binary representation at process 150. Since certain commands, e.g. commands that merely display information, do not change the state of the system, user confirmation is not required. There is no security problem in not confirming these commands since the system only allows users to view information to which they have access.

There are certain commands that change the state of the system may not require confirmation. Some commands may allow the user to turn on a file to receive the user's redirected output. This files are sometimes known in the art as "listing files." Generally, the turning on of a listing file is a confirmed command.

In situations where a listing file has been turned on (i.e., the output is redirected into this file) commands which normally display information may, by writing to the listing file, alter the state of the system. In these cases, confirmation of these commands is generally not required because the flow of information is constrained by the TCB's security rules.

When a command requesting the display of information is executed by the TCB 10, the SSVR 12 responds by displaying a complete representation of the requested command as well as the requested information, i.e. the standard human-readable representation of the command. In one embodiment, the SSVR 12 displays

the function code, command modifiers, and command parameters all in unambiguous notation. If employed, the process ID may also be displayed.

Thus, if the user's command is improperly parsed, or modified in an unauthorized manner, the user will be able to observe the difference. This inhibits an untrusted subject from fooling a user into believing that he or she is observing the results of a submitted command, when in fact that command had actually been altered.

If the command requested is one for which confirmation is required, the SSVR 12 displays a standard human-readable representation of what is about to be done to the user terminal at process 170. The code representing each command should produce a different display, unless the commands have the same effect. As previously discussed, the re-display of the action that a binary representation will perform is less complex than the parsing of the original command. This is in part because by definition there is only one standard human-readable representation for each parsed command.

By parsing the command in untrusted code, and verifying the request in trusted code, the overall complexity of the TCB 10 is decreased. The display of "what is about to be done" may of course vary from one command to another.

If the command's purpose is to add a record to a database or create an object, the display may contain what the new record will contain, or what the attributes of the object will be if the command is confirmed.

If a particular database or object is to be modified, the display may show what the updated data will look like, or what the attributes of the object will be if the command is confirmed. For commands which remove an existing object, or delete an existing file, the display may show the record or object.

Generally, for both adding/creating and modifying, the data packet representing the submitted command may contain fields for the objects attributes, such as its name, its size and its protection. For each of these fields, the packet may also have a flag indicating whether the field is meaningful. These flags indicating meaningful fields may be know as "field selectors."

In add/create commands, the TCB may supply default values for unselected fields. In commands which modify a database entry or object, unselected fields may retain their previous values.

To aid the user, the selected fields may be highlighted when displayed for confirmation. Such highlighting may be accomplished by methods known in the art (e.g., by using escape sequences that begin and end reverse video). This highlighting draws the users attention to the fields that have non-default or changed values and helps to protect the user from tricks by untrusted code.

When an audit record is produced of the confirmation display via the sink (discussed below) the reverse-video escape sequences may be replaced with dashes on the next line. Example:

Name: JOHN.DOE
- - - -

This change permits the audit record to be properly displayed on a wide variety of devices, even simple line printers.

Alternative embodiments are envisioned in which the TCB checks each field's value: for add/create commands the fields that have non-default values are highlighted; and for modify commands the fields that are changing are highlighted. This "field comparison" approach generally results in less highlighting.

Generally, for every command, every packet field is displayed on the screen, except for fields ignored by the particular command.

In the cases when a database will not be modified, the display generated by the SSVR 12 may express what action is to be done. In this manner, the possibility of an erroneous confirmation is reduced since the user is requested to confirm the results, not the command.

The SSVR 12 may also display a list of user protection attributes so that the user knows what role he or she is exercising. The user is required to confirm explicitly (Yes or No) the action requested. In a general embodiment, the TCB 12 displays for confirmation the results which would occur if the command is executed.

Generally, any information in the binary representation that is displayed is first validated. This information is checked once to ensure that certain requirements are met, e.g., to ensure that the counts of array elements are within array bounds (since out-of-bounds conditions could provide an opportunity for entry into the TCB by illicit software).

Before displaying the contents of a packet containing the binary representation of the command, anything that could cause the formatting code to malfunction may be validated. Generally, such candidates for validation could include anything used to index into something else. Specific examples include the PASCAL language's implementation of varying strings (a count followed by an array of characters; the count is checked to be less than or equal to the character array size), and numbers that represent keywords (their value is checked before it is used to index into an array of keyword text strings). Further methods of validation are well known in the art and will not be discussed herein.

In one embodiment, the replies that the user may make to the confirmation prompt are "yes", "no", Ctrl-C, Ctrl-Y, or the BREAK key. No default responses exist in this embodiment. For each user response, the secure server may display a message restating the response. If the response was "yes" or "no", the secure server 12 displays an indication of the computing session in the UOS 20 when it is being be resumed after command completion.

If, however, the user activates the Ctrl-Y, Ctrl-C, or the BREAK key, the command will be aborted. In one embodiment, the CC then changes the command status code to negative. This may be done to inform the submitting process that the command has been aborted. In these cases a direct session with the SSVR 12 will be established at process 300. In these instances, the user must explicitly issue a resume command to return to the session in the UOS 20.

If the user confirms the results or command, by replying with "yes", or if the command does not require confirmation, the TCB 10 enforces the system's security policy at process 145. At this process the TCB 10 determines if the user has the necessary access privileges to perform the requested activity. If the user has the required access rights, the SSVR 12, along with other code in the TCB 10 attempts to execute the command in a conventional manner.

Upon attempted execution of the command, the SSVR 12 may inform the CC 14 of the execution status. In some high-security situations, concern for information flow from trusted to untrusted code will prevent informing the CC 14. The CC 14 may change the command status to indicate the SSVR 12 response. In addition, in alternate embodiments, the CC 14 changes the command state from command in progress 420 (FIG. 7) to command done 430. In these cases, the CC 14 notifies the submitting untrusted subject by traditional methods (e.g., by advancing an event count for that process). The SSVR 12 then reestablishes the terminal connection with the computing session in the UOS 20 from which the command was executed.

In embodiments utilizing the status code, if the user responds with "no", the secure server informs the session in the UOS 20 that the command was not executed (negative status code). If command states are also employed, the SSVR 12 changes the command state to command done, and re-establishes the user's session in the general UOS 20 at process 160.

If the trusted command requested by the user is a command which affects one of the system databases, and the database record was changed by some other process or user between the time the confirmation display appeared and the current user confirmed the command, then the SSVR 12 displays a message indicating that a change has been made and requests reconfirmation. In this case, the SSVR 12 waits for a response as if it were the first time the information was displayed.

To ensure that if two users attempt to modify the same element at the same time, only one request is rejected, the TCB may utilize identifiers known as "modification counts" (mod-counts).

Each element of key databases (or registries) may contain a modification count. This mod-count is advanced after each modification and thus identifies how many modifications have been made to the database element. Thus, when a user attempts to modify a database element, the SSVR 12 first obtains that database element's mod-count. After the user has confirmed the request, the TCB 10 checks the mod-count in the request against the current mod-count.

A different mod-count indicates that a change has been made to the database element, and the user is asked to reconfirm the request since the result previously confirmed by the user is not what the result would actually be.

To deter malicious users or subjects from preventing the execution of trusted commands, two measures may be taken. First, certain access rights may be required to change certain database elements. Second, the mod-count should not be advanced when a modification is requested that does not change the database element. In this manner, attempts by malicious subjects to prevent execution of a trusted command, by rapidly and spuriously causing the mod-count to be incremented, will be inhibited.

Once activity in the SSVR 12 is completed, the user's session in the general UOS 20 is reestablished. In one embodiment, this may be accompanied by the CC 14 informing the untrusted subject of the status of the submitted command. The untrusted code in the general UOS 20 then wakes-up the untrusted process running in the UOS 20 at process 160. In one embodiment, before the TCB 10 passes the user to the untrusted subject, the SSVR 12 prompts the user to activate a carriage return. This step is necessary to inform the user that he/she is being placed into an untrusted environment.

4.10 The SSVR as an Audit Recorder

In one embodiment, the SSVR 12 audits (i.e., creates an audit trail by writing to an audit or log file) at the command level. For direct commands the SSVR 12

audits both the command line and the final status of the command. For trusted commands (secure-attention commands), the SSVR 12 audits both the confirmation display and the final status of the command.

To record the confirmation displays, the SSVR 12 defines a portion of memory to store the strings of information sent by the SSVR 12 to the user's terminal. In one embodiment, this portion of memory is known as the "sink." Only the strings that comprise the SSVR's confirmation screens are placed in the sink.

Strings may be appended to the sink only when the sink is opened. The sink is opened when the SSVR 12 begins to service a trusted command. The sink may be closed after the user has responded to the confirmation prompt, or before the display of certain requested information. The information stored in the sink does not include the user input to any prompts.

The information stored in the sink is maintained for auditing purposes and may be utilized according to traditional auditing practices.

4.11 Confirmation by Others than the Submitting User

In some situations it may be desirable for users other than the submitting user to confirm a secure-attention command. Such a situation may arise if there is a certain application or command file that includes a privileged command but is needed to be run by a large number of users. In the cases where it may be impracticable to grant all of the users the necessary access rights, it may be possible to grant one user the neces-

sary rights, and to allow him/her to confirm this command for the other users.

Such non-submitting user confirmation may be accomplished by having the code in the untrusted subject lie about which user is submitting the command. For example, the code in the UOS 20 may ensure that certain commands are always sent to the TCB 10 as if they had been submitted by a selected user.

One purpose of this select user would be to confirm these certain commands for the other users. After a non-select user submits one of the certain commands, the selected user will be notified that there is a submitted command waiting. The select user may then observe the confirmation display. If the select user determines that the command submitted was a legitimate command, he/she may confirm the command. Otherwise, the original users attempt to execute the command fails.

4.12 Implementation Approaches

The method for executing trusted commands described above may be implemented through the use of appropriate software. The actual design and coding is a matter of routine for those skilled in the art having the benefit of this disclosure. Such a design will of course vary with the hardware platform and other factors associated with the specific desired implementation.

FIG.5 illustrates one such method for implementing the present invention. A programmable machine comprises a central processing unit (CPU) 210, and an input/output (I/O) device 220. The I/O unit 220 is

capable of reading and transmitting stored programming instructions from known storage media 230. The CPU 210 is capable of processing these instructions and executing the programming steps represented on media 230. Such systems are well known in the art and will not further be described.

To implement the invention, the necessary programming steps are stored on storage media 230 by traditional methods. The storage media 230 is then read by the I/O unit 220 and the programming instructions are transmitted to the CPU 210. The CPU 210 reads and interprets the instructions, and may thereafter carry out the method of the invention.

CLAIMS

WHAT IS CLAIMED IS:

1. A machine-executable method for executing a trusted command issued by a user, said method comprising the steps of:

   (a) parsing the trusted command in an untrusted computing environment to generate a parsed command;

   (b) submitting the parsed command to a trusted computing environment; and

   (c) executing the parsed command in the trusted computing environment.

2. A method including the steps of claim 1 and additionally including the steps, executed after step (b) of claim 1, of:

   (1) in the trusted environment, displaying a representation of the parsed command to the user;

   (2) receiving a signal from the user signifying whether the displayed representation accurately represents the user's intentions;

   (3) if the signal signifies that the displayed representation does not accurately represent the user's intentions, then preventing the performance of step (c) of claim 1.

3. The method of claim 2 wherein the representation of the parsed command is displayed, and the signal from the user is received, through a trusted path.

4. The method of claim 1 wherein the trusted computing environment comprises a security kernel.

5. The method of claim 1 wherein the untrusted computing environment comprises a general operating system.

6. A method for executing in a computing system a trusted command issued by a user, said method comprising the steps of:

   (a) receiving user identification data from the user via a trusted path;

   (b) receiving the trusted command from the user via an untrusted path;

   (c) parsing the trusted command in an untrusted computing environment to generate a parsed command;

   (d) submitting the parsed command to a trusted computing environment;

   (e) in the trusted computing environment, performing a security check on the parsed command and user identification data; and

   (f) in the trusted computing environment, executing the trusted command.

7. The method of claim 6, wherein the security check enforces an Orange Book security criterion.

8. A method including the steps of claim 6 and additionally including the steps, executed after step (d) and before step (f) of claim 6, of:

(1) in the trusted environment, displaying a representation of the parsed command to the user;

(2) receiving a signal from the user signifying whether the displayed representation accurately represents the trusted command; and

(3) if the signal signifies that the displayed representation does not accurately represent the trusted command, then preventing the performance of step (f) of claim 6.

9. A method including the steps of claim 6 and additionally including the steps, executed after step (d) and before step (f) of claim 6, of:

(1) in the trusted environment, displaying a representation of the parsed command to a second user;

(2) receiving a signal from the second user signifying whether the displayed representation accurately represents a legitimate command; and

(3) if the signal signifies that the displayed representation does not accurately represent a legitimate command, then preventing the performance of step (f) of claim 6.

10. A method for ensuring the existence of a trusted path in a computing system comprising the steps of:

(a) in a trusted computing environment, upon login by a user, assigning a process identifier to the user in the trusted computing environment;

(b) storing the assigned process identifier in trusted memory;

(c) establishing a trusted path;

(d) in the trusted path, displaying the process identifier to the user; and

(e) upon a subsequent entry into the trusted path, displaying the process identifier to the user.

11. The method of claim 10 wherein the process identifier is a randomly or pseudo-randomly generated group of alphanumeric characters.

12. The method of claim 11 wherein the process identifier is pronounceable.

13. An automatic data processing machine programmed to execute the method of any one of claims 1 to 12.

14. An automatic data processing machine comprising means for performing the method steps of any one of claims 1 to 12.

15. A program storage device readable by a machine and tangibly embodying a representation of a program of instructions adaptable to be executed by said machine to perform the method of any one of claims 1 to 12.

16. Apparatus for executing a trusted command that is issued by a user and that is parsed by untrusted parsing means to generate a parsed command, comprising:

(a) trusted means for receiving the parsed command; and

(b) trusted means for executing the parsed command.

17. Apparatus for controlling the execution by a machine of a trusted command that is issued by a user and that is parsed by untrusted parsing means to generate a parsed command, comprising;

(a) trusted-program storage means, readable by the machine, for causing the machine to receive the parsed command from the untrusted parsing means; and

(b) trusted-program storage means, readable by the machine, for causing the machine to execute the parsed command.

18. Apparatus for controlling the execution by a machine of a trusted command that is issued by a user with user identification data and that is parsed by untrusted parsing means to generate a parsed command, comprising;

(a) trusted program storage means, readable by the machine, for causing the machine to receive the user identification data from the user;

(b) trusted program storage means, readable by the machine, for causing the machine to receive the parsed command from the untrusted parsing means;

(c) trusted program storage means, readable by the machine, for causing the machine to perform a security check on the parsed command and a security check on the user identification data; and

(d) trusted program storage means, readable by the machine, for causing the machine to execute the trusted command.

19. Apparatus as in claim 18 and additionally comprising:

(1) trusted program storage means, readable by the machine, for causing the machine to display a representation of the parsed command to the user;

(2) trusted program storage means, readable by the machine, for causing the machine to receive a signal from the user signifying whether the displayed representation accurately represents the trusted command; and

(3) trusted program storage means, readable by the machine, for preventing the machine from executing the trusted command if the signal signifies that the parsed command does not accurately represent the trusted command.

20. Apparatus as in claim 18 and additionally comprising:

(1) trusted program storage means, readable by the machine, for causing the machine to display a representation of the parsed command to a second user;

(2) trusted program storage means, readable by the machine, for causing the machine to receive a signal from the second user signifying whether the displayed representation accurately represents a legitimate command; and

(3) trusted program storage means, readable by the machine, for preventing the machine from executing the trusted command if the signal signifies that the parsed command does not accurately represent a legitimate command.

ABSTRACT

A method for executing trusted commands, in which a trusted command is first received from a user at a user terminal and parsed by untrusted code; then passed to a trusted computing base for execution. The trusted computing base displays some indication of what is to be done back to the user for confirmation. Confirmation of the commands prevents unauthorized modification of the commands and increases system confidence. A randomly (or pseudo-randomly) generated process identifier is employed to verify the existence of a trusted path.

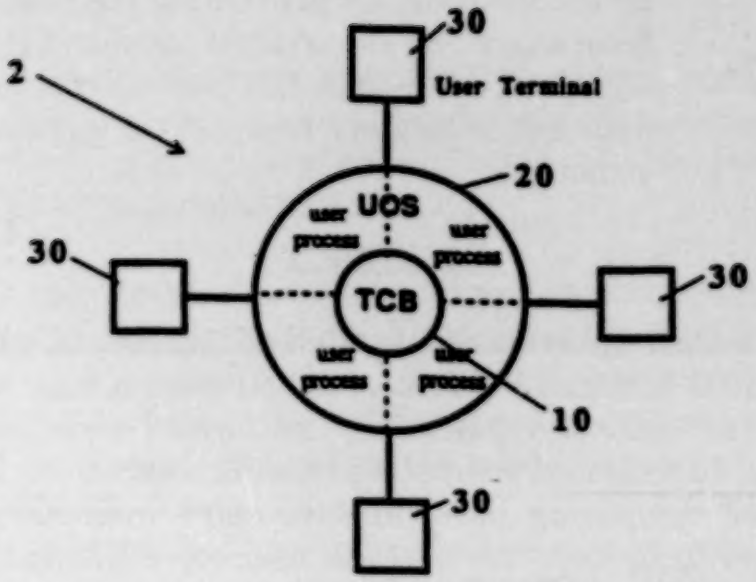

Figure 1

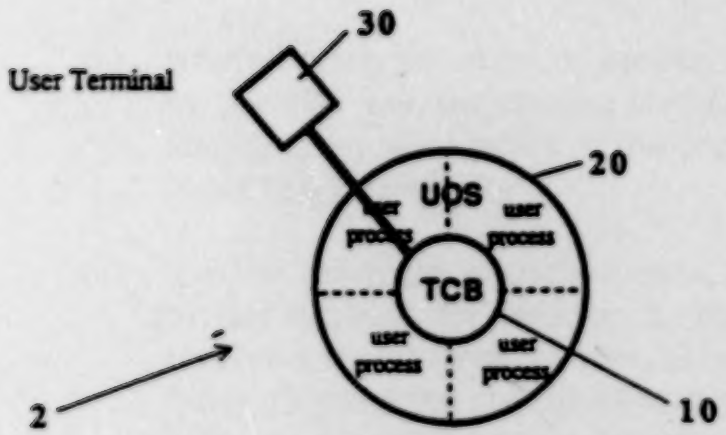

Figure 3

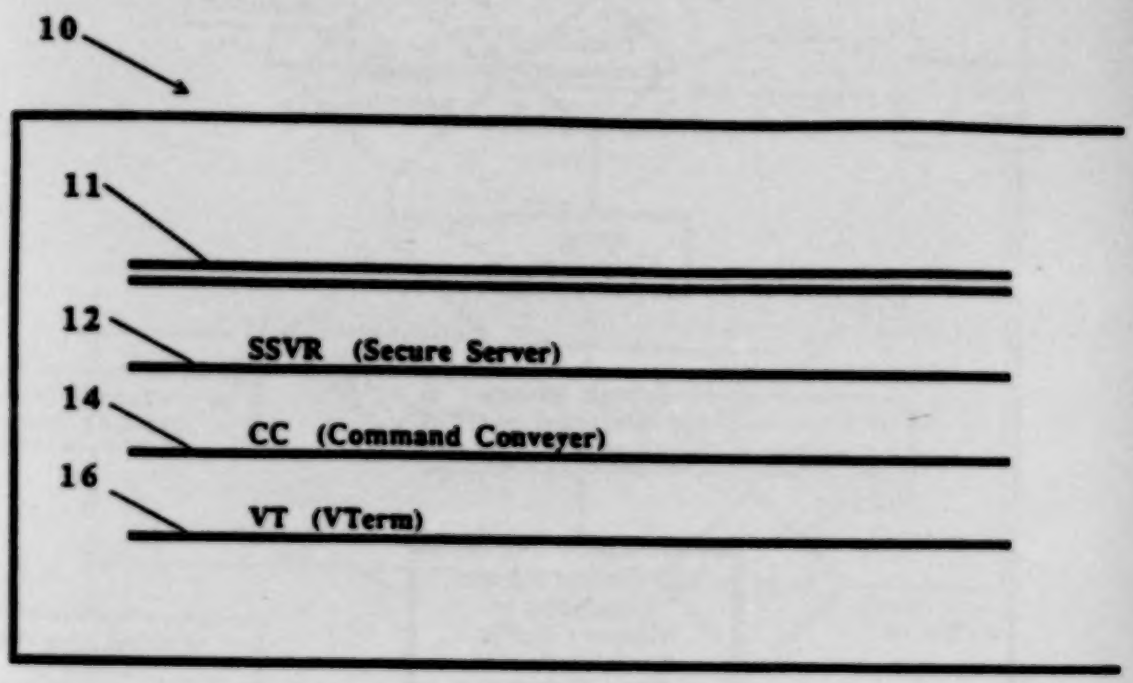

Figure 2

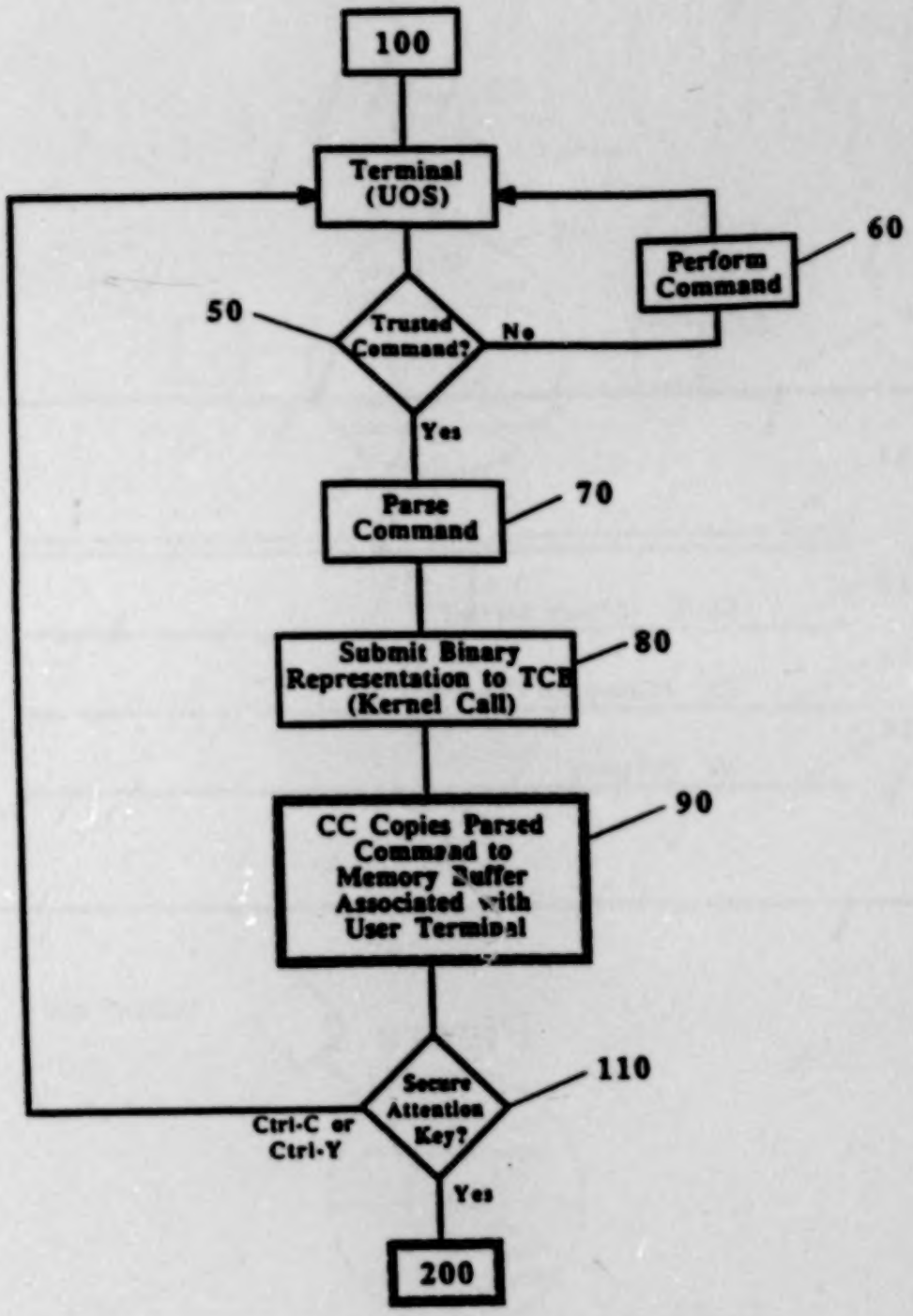

Figure 4A

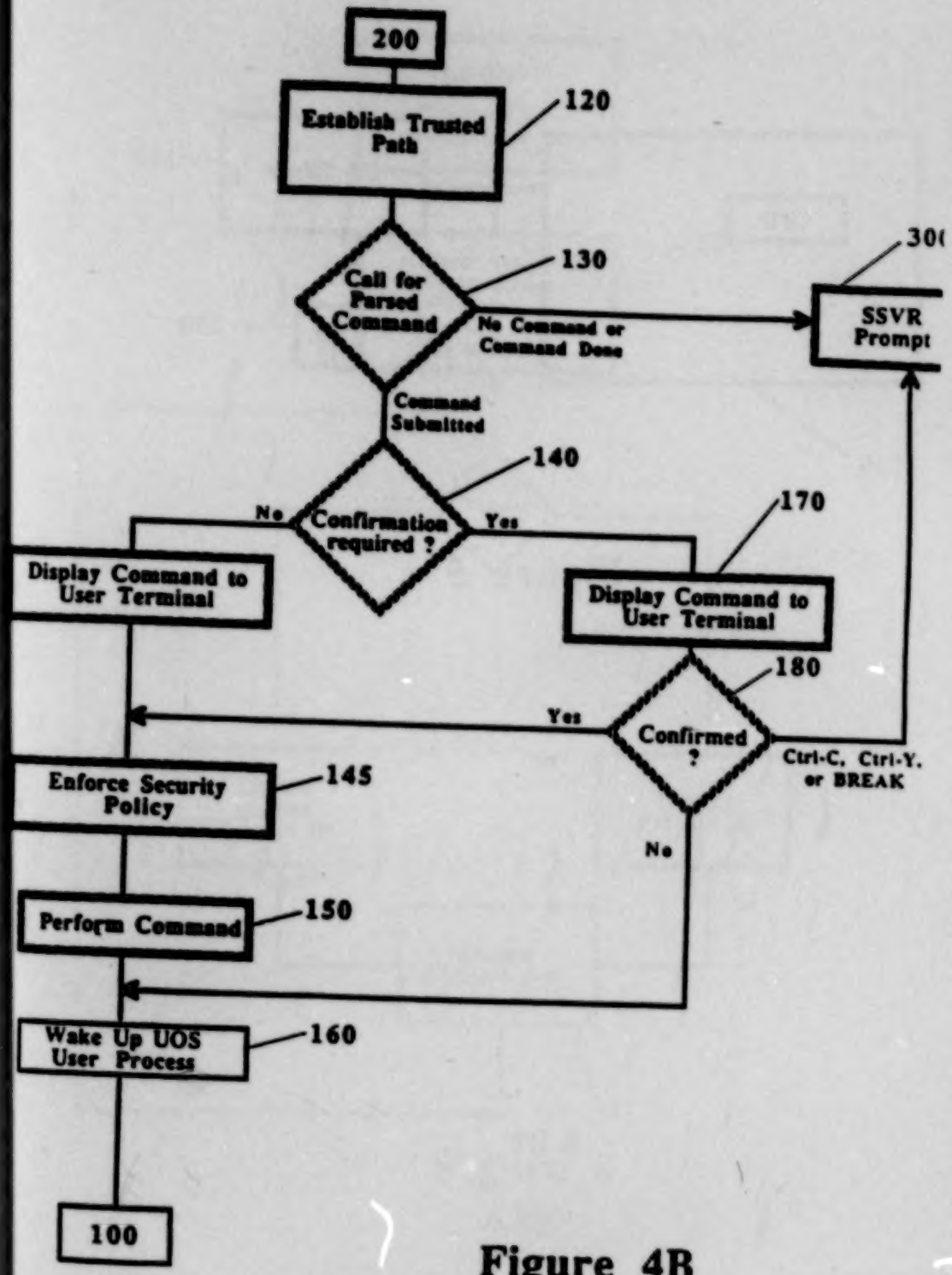

Figure 4B

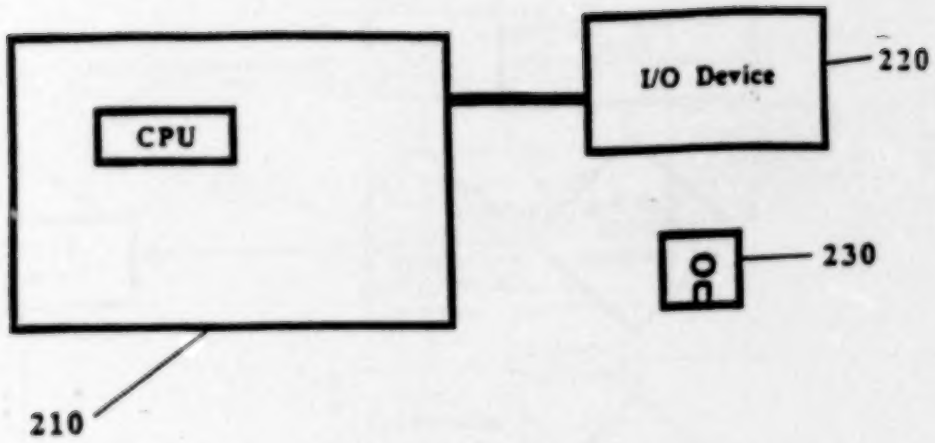

Figure 5

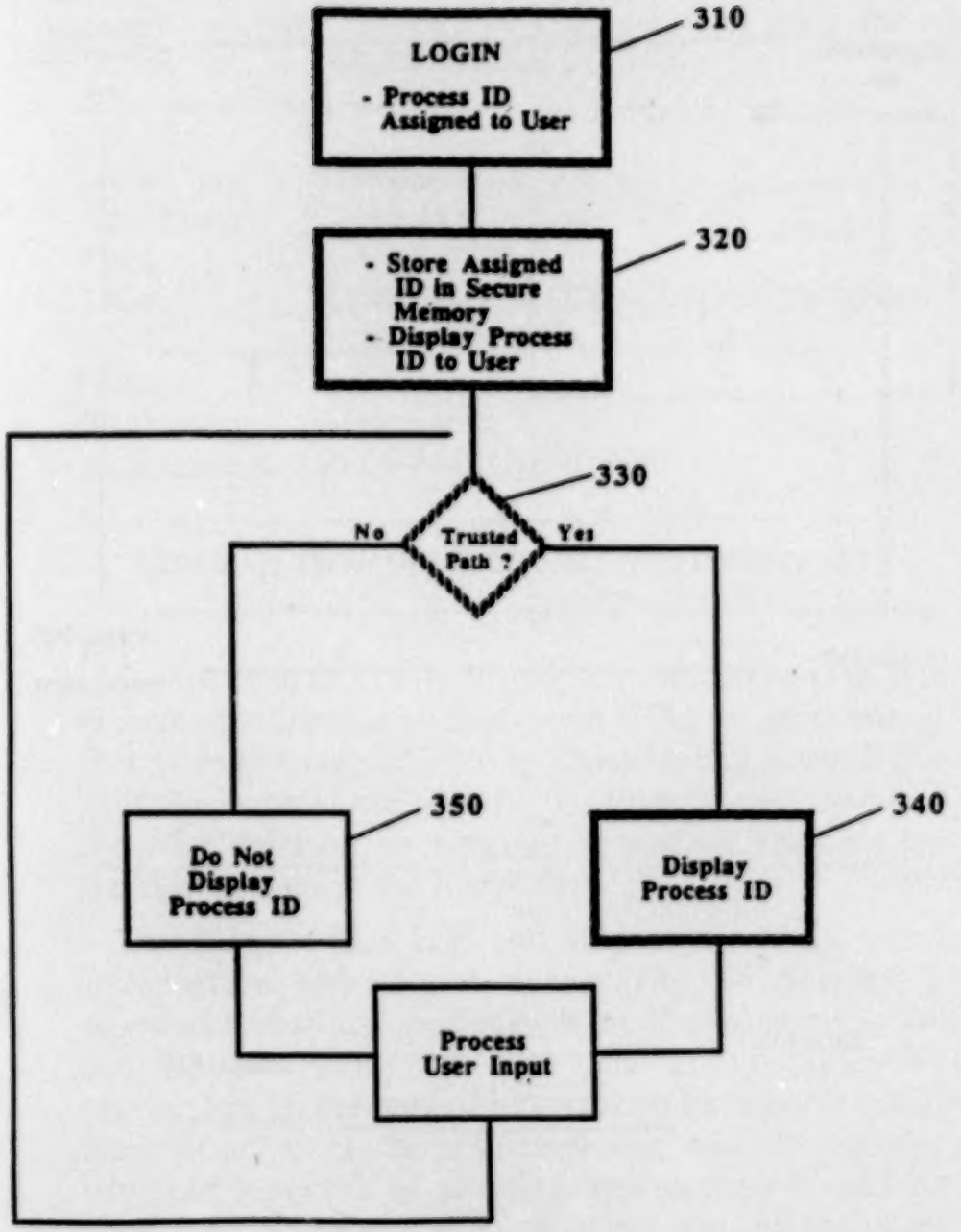

Figure 6

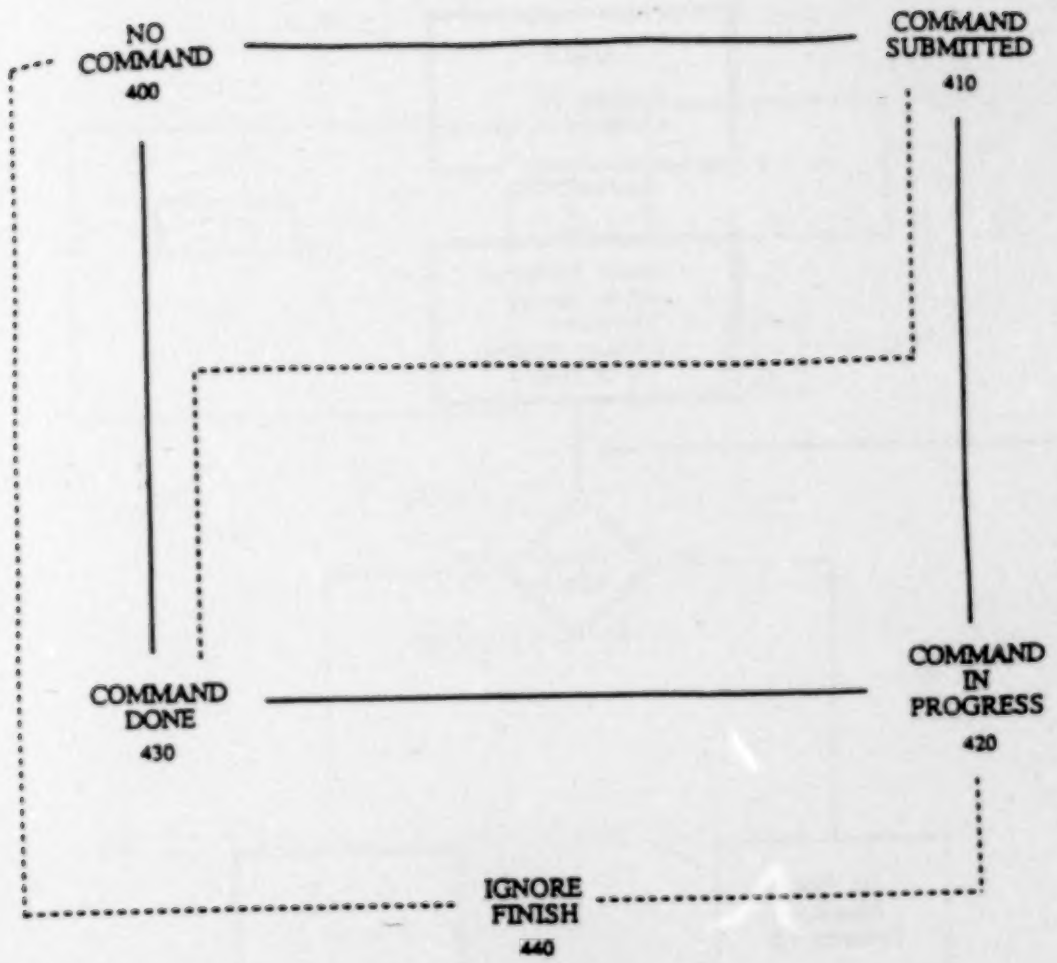

FIGURE 7

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.:	479,666
Applicant:	Zurko et al
Filed:	Feb. 13, 1990
Title:	METHOD FOR EXECUTING TRUSTED-PATH COMMANDS
Group:	234
Examiner:	(Unknown)
Atty docket:	PD90-0141 (DIGU:012)

INFORMATION DISCLOSURE STATEMENT (#2)

Pursuant to 37 C.F.R. § 1.56, it is requested that this Information Disclosure Statement (IDS) be entered and that the information listed on the attached Form PTO-1449 be considered by the Examiner and made of record. Copies of the cited references are enclosed for the convenience of the Examiner.

The filing of this IDS and submission of the cited information should not necessarily be deemed a concession that any particular item of cited information constitutes "prior art" to the above-captioned application. In the case of information not clearly qualifying as prior art, the Examiner may wish to consider whether a search of pending applications would be appropriate to determine whether any of the individuals identified in such information have filed pending applications.

In accordance with 37 C.F.R. § 1.97(b), this Information Disclosure Statement is not to be construed as a representation that a particular search has been made nor that other possibly-material information as defined in 37 C.F.R. § 1.56(a) does not exist.

The comments below are believed to be a concise summary of the pertinence of the cited information. The comments below cannot and do not, of course, take the place of the Examiner's own careful consideration of each item of cited information. The references are grouped as follows for purposes of discussion, and are discussed within each group in alphabetical order by inventor/author:

Group 1 References regarded as being pertinent and/or comprising useful general background information;

Group 2 References regarded as being of general interest;

Group 3 References regarded as being of at most peripheral interest (e.g., references obtained by the undersigned attorney because they were themselves cited in another reference of potential interest).

GROUP I REFERENCES

Dunford, "FILER (Version 2.20)" User Documentation describes a FILER program that has been available on the publicly accessible CompuServe information service (and in fact in use on the undersigned attorney's computer system) since before one year prior to the filing date of the application.

The FILER program repeats back potentially dangerous user commands and requests confirmation from the user prior to execution. For example, a user may instruct the FILER program to erase a group of files using "wild card" designators (e.g., "DEL*.*"). Prior to executing the command, the program displays a list of the specific files matching the wild card designator that will be erased and asks for confirmation that the files should be deleted. The /S command-line option of the program permits the user to confirm selectively the deletion of individual files matching the wild card designator. See page 1, lines 24-25, 35-37; page 5, lines 195-201; page 6, lines 255-258.

UNIX operating system (no specific reference included): In many UNIX systems, a number of simple shell commands translate directly into system calls. This feature is well documented. For example, the "rm" command corresponds to the "un-link" system call and results in removal of a file. The "rmdir" command corresponds to the "rmdir" system call and results in removal of a directory. The "chmod" command corresponds to the "chmod" system call and results in a change to the protection mode of a file.

These system calls could be regarded as "trusted." A UNIX system call does its work in kernel mode. The call does something that an unprivileged program running in an unprivileged account could not do directly. For example, an unprivileged program running in an unprivileged account cannot write to the file structure. Yet the system calls cited above can all be used by unprivileged programs running on behalf of unprivileged users. Such programs may be deemed "untrusted". The "rm", "rmdir", and "chmod" programs themselves are unprivileged on many UNIX-like

system such as an Ultrix system developed by the assignee, Digital Equipment Corporation.

Thus, the prior art includes an untrusted program parsing a command and then executing the command by a calling a trusted service that executes in a trusted computing environment, namely kernel mode. This art is believed to be pertinent to claim 1 prior to its amendment.

Johri et al. '653 purposes a trusted-path mechanism for an operating system to inhibit Trojan-horse programs. By pressing a Secure Attention key (SAK) on a terminal keyboard, a user may create a trusted path between the terminal and a trusted-shell portion of a trusted computing base (TCB) in a UNIX-type operating system. *See* col. 2, lines 55-70. When a SAK signal is received by the host, the host terminates all processes in the terminal's process group; removes essentially all addressing information relating to the terminal's device-driver interfaces from the data processing system; and executes a new child process and a trusted shell process. *See* col. 3, lines 10-53.

McCauley & Drongowski, *KSOS; The Design of a Secure Operating System,* was published in (it is believed) about 1979. The paper discusses a DOD Kernalized Secure Operating System (KSOS) that apparently was under development at the time. Section 3.1 discusses the security kernel concept; section 3.1.1 includes a background discussion of trusted processes.

Section 3.1.5, beginning at page 9, discusses a proposed secure terminal interface, including the concept of blocking the normal path when the terminal user presses a reserved attention character such as the

BREAK key. Figure 2 of the paper illustrates the proposal.

The paper includes a bibliography of references beginning at page 14.

* * * * *

[Group II and III references omitted]

Date: 5-1-91 /s/ D. C. TOEDT
D.C. TOEDT
Reg. No. 31,144

ARNOLD, WHITE & DURKEE
750 Bering Drive (77057)
P.O. Box 4433
Houston, Texas 77210-4433
713-787-1400 (voice)
713-789-2679 (FAX)

ATTORNEYS FOR ASSIGNEE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Mary Ellen Zurko et al.

Serial No.: 479,666 Group Art Unit:

Filed: February 13, 1990 Examiner:

Title: METHOD FOR EXECUTING TRUSTED-PATH COMMANDS

Atty. Dkt: PD90-0141
(DIGU: 012)

PRELIMINARY AMENDMENT AND SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Hon. Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Sir:

Prior to the receipt of the initial Office Action, please amend the above referenced application as follows:

IN THE SPECIFICATION:

Page 6, line 21, after "is usually operated", insert the phrase—in conjunction with—.

Page 7, line 1, after "the majority of", insert—the—.

Page 7, line 10, after "and either", insert—(1)—.

Page 7, line 11, after "its own initiative", insert—(2)—.

Page 8, line 7, replace "to have" with—that are given—.

Page 8, line 19, replace "Because" with—Partly because—.

Page 15, line 21, change the phrase "command line" to the phrase—command-line—.

Page 16, line 12, replace "is" with—are—.

Page 17, line 24, after "VMS or Ultrix system", insert — (the UOS) —.

Page 18, line 17, change the word "VTerm" to the phrase—Virtual Terminals—.

Page 19, line 4, delete "a".

Page 20, line 13, delete "user".

Page 23, line 1, replace "the terminal" with—certain terminals—.

Page 27, lines 6-7, replace "as long as the user has an active session" with—until the user logs out of the TCB 10—.

Page 30, line 7, replace "their" with—his/her—.

Page 33, line 3, delete "a".

Page 35, line 4, change "errors" to—inaccuracies—.

Page 35, line 8, after "submitting to the TCB", please insert— the parsed representation of the command and —.

Page 35, line 10, before "virtual terminal", insert—identification of the—.

Page 36, line 8, after "command has been", insert the word—successfully—.

Page 36, line 17, delete "and".

Page 36a, line 22, replace ";" with—,—.

Page 38, line 3, delete "the user's process".

Page 38, line 5, delete "activities" and insert therefore—disruptions—.

Page 41, line 2, delete "determine", and insert therefore—determines—.

Page 41, line 8, delete "only".

Page 41, line 9, before "information", insert—only—.

Page 41, line 11, after "system", insert—which—.

Page 41, line 13, replace "turn on a file to receive the user's redirected output" with —direct an informational display into a file to which the user has access permission—.

Page 41, line 13, delete "This", and substitute therefore—These—.

Page 41, line 18, delete "this file" and insert therefore —a listing file—.

Page 41, line 22, change "rules" to—policy—.

Page 42, line 13, change "had" to—has—.

Page 43, line 19, change "objects" to—object's—.

Page 44, line 8, change "users" to—user's—.

Page 44, line 12, replace "is produced of the confirmation display" with—of the confirmation display is produced—.

Page 45, line 8, after "may express", please insert—, in a human language (e.g., English), —.

Page 45, line 10, delete ", not the command", and insert therefore—(an understandably stated action), not a potentially cryptic command—.

Page 45, line 12, after "protection attributes", insert therefore—(such as the privileges the user has enabled) —.

Page 47, line 20, change "e.g.," to—or —.

Page 48, line 24, delete "element".

Page 49, line 23, change "wakes-up" to—wakes up—.

Page 52, line 7, change "users" to—user's—.

IN THE CLAIMS

Please amend claim 1, without prejudice, to read:

1. (once amended) A machine-executable method for executing a trusted command issued by a user, said method comprising the steps of:

   (a) parsing the trusted command in an untrusted computing environment to generate a parsed command:

   (b) submitting the parsed command to a trusted computing environment; [and]

   (c) in the trusted environment, displaying a representation of the parsed command to the user;

   (d) receiving a signal from the user signifying whether the displayed representation accurately represents the user's intentions;

   (e) if the signal signifies that the displayed representation does not accurately represent the

user's intentions, then preventing the execution of the parsed command.

[(c) executing the parsed command in the trusted computing environment.]

Please amend claim 3, without prejudice, to read:

3. (once amended) The method of claim [2] 1 wherein the presentation of the parsed command is displayed, and the signal from the user is received, through a trusted path.

Please amend claim 16, without prejudice, to read:

16. (once amended) Apparatus for executing a trusted command that is issued by a user and that is parsed by untrusted parsing means to generate a parsed command, comprising:

(a) trusted means for receiving the parsed command via a trusted path; [and]

(b) means for displaying a representation of the parsed command to the user; and

([b] c) trusted means for executing the parsed command.

IN THE ABSTRACT

Page 65, line 6, after "displays", insert —to the user for confirmation—.

Page 65, line 7, delete "back to the user for confirmation."

REMARKS

This amendment is made before the receipt of the initial Office Action. The Examiner's attention is invited to the following amendments:

(a) page 35, line 8: the amendment makes it clear that the parsed representation of the command, as well as the terminal identification, is passed to the TCB. The Examiner will recognize that this amendment is merely editorial, inasmuch as the substance of the change is fully supported elsewhere (e.g., in Figure 4A at process 80 and at page 13, lines 8-11).

(b) page 47, line 20: "e.g.," is changed to—or—because event counts are not regarded as conventional means in this context. The Examiner will recognize that the change is not necessary for enablement or best-mode support of the claims and does not constitute the introduction of new matter.

Otherwise, the changes to the specification are requested to correct minor typographical errors and to increase the readability of the application. The Examiner will recognize that no new matter has been introduced.

The claims have been amended to point out more particularly and distinctly claim the subject matter which the applicants regard as their invention. It is believed that claims 1 and 16 may be invalid as overly broad in view of prior art that may exist (the applicants are attempting to obtain more specific information about whether or not this in fact is the case).

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

[Omitted]

* * * * *

The claims are believed to be in condition for allowance. Early passage to issue is requested.

/s/ D.C. TOEDT V 5-1-91
D.C. Toedt III
Reg. No. 31,144

ARNOLD, WHITE & DURKEE
P.O. Box 4433
Houston, Texas 77210
713-787-1400 (voice)
713-789-2679 (FAX)

ATTORNEYS FOR APPLICANTS/ASSIGNEE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
Mary Ellen Zurko, et al. Group Art Unit: 2307
Serial No.: 479,666 Examiner: Brian Klock
Att'y Docket: PD90-0141
(DIGU:012)

Filed: February 13, 1990

For: METHOD FOR EXECUTING TRUST-PATH COMMANDS

RESPONSE TO OFFICE ACTION OF MARCH 18, 1992

Honorable Commissioner
of Patents and Trademarks
Washington, D.C. 20231

Sir:

In response to Office Action mailed March 18, 1992, please amend the referenced application as follows:

In the Claims:

Please amend claim 1 to read:

1. (twice amended) A machine-executed method for executing a trusted command issued by a user on a computer system, the computer system including an untrusted computing environment and a trusted computing environment, said method comprising the steps of:
   (a) parsing the trusted command in [an] the untrusted computing environment to generate a parsed command;

(b) submitting the parsed command to [a] the trusted computing environment;

(c) [in the trusted environment,] displaying a representation of the parsed command to the user through a trusted path;

(d) receiving a signal from the user through a trusted path signifying whether the displayed representation accurately represents the user's intentions;

(e) if the signal signifies that the displayed representation does not accurately represent the user's intentions, then preventing the execution of the parsed command; and

(f) if the signal signifies that the displayed representation accurately represents the user's intentions, executing the parsed command in the trusted environment.

Please cancel claim 2 without prejudice as being encompassed in claim 1.

Please cancel claim 3 without prejudice as being encompassed in claim 1.

Please amend claim 6 to read:

6. (once amended) A machine-executed method for executing [in a computing system] a trusted command issued by a user on a computing system including an untrusted computing environment and a trusted computing environment, said method comprising the steps of:

(a) receiving user identification data in the trusted computing environment from the user via a trusted path;

(b) receiving the trusted command from the user in the untrusted computing environment via an untrusted path;

(c) parsing the trusted command in [an] the untrusted computing environment to generate a parsed command;

(d) submitting the parsed command to [a] the trusted computing environment;

(e) [in the trusted computing environment,] performing a security check on the parsed command and user identification data in the trusted computing environment; and

(f) [in the trusted computing environment,] executing the trusted command in the trusted computer environment.

Please amend claim 7 to read:

7. The method of claim 6, wherein the security check enforces [an Orange Book] *a* security criterion from the Department of Defense Trusted Computer System Evaluation Criteria (Ref. No. DOD 5200.28-STD).

Please amend claim 10 to read:

10. A machine-executed method for [ensuring] verifying the existence of a trusted path to a user in a computing system, the computer system including a trusted computing environment, the method comprising the steps of:

(a) [in a trusted computing environment,] upon login by a user, assigning a process identifier to the user in the trusted computing environment;

(b) storing the assigned process identifier in trusted memory;

(c) establishing a trusted path between the user and the trusted computing environment;

(d) [in] through the trusted path, displaying the process identifier to the user; and

(e) upon [a] the user's subsequent entry into the trusted [path] environment, displaying the process identifier to the user through the trusted path.

Please amend claim 16 to read:

16. (twice amended) Apparatus for executing a trusted command that is issued by a user and that is parsed by untrusted parsing means to generate a parsed command, comprising:

(a) trusted means for receiving the parsed command via a trusted path;

(b) means for displaying a representation of the parsed command to the user for verification; and

(c) trusted means for executing the verified parsed command.

Please cancel claims 17-20 without prejudice to possible representation at a later date.

Please substitute the following claim for claim 15:

21. (new) A program storage device readable by a machine and encoding machine-executable instructions that include instructions for performing the methods of any one of claims 1 though 12.

REMARKS

[Omitted]

* * * * *

Respectfully submitted,

/s/ D. C. TOEDT III
D. C. TOEDT III
Reg. No. 31,144

ARNOLD, WHITE & DURKEE
P.O. Box 4433
Houston TX 77210-4433
713-787-1400 (AWD)
713-787-1408 (DCT direct)

ATTORNEYS FOR ASSIGNEE

IN THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

Appeal No. 96-1258
(Serial No. 07/479, 666)

IN RE MARY E. ZURKO, THOMAS A. CASEY, JR., MORRIE GASSER, JUDITH S. HALL, CLIFFORD E. KAHN, ANDREW H. MASON, PAUL D. SAWYER, LESLIE R. KENDALL AND STEVEN B. LIPNER, APPELLANTS

APPEAL FROM A DECISION OF THE
BOARD OF PATENT APPEALS AND
INTERFERENCES DATED JULY 31, 1995

[Filed: June 20, 1997]

APPELLANTS' OPPOSITION TO CORRECTED PETITION FOR REHEARING AND SUGGESTION FOR REHEARING *IN BANC* BY APPELLEE

[List of counsel and certificate of
interested parties omitted]

TABLE OF CONTENTS

		Page
INTRODUCTION		[90]
I.	THE PETITION IMPROPERLY RAISES ISSUES FOR THE FIRST TIME ON REHEARING	[90]
II.	THE PETITION DOES NOT DEMONSTRATE THAT THIS COURT'S DECISION WOULD HAVE BEEN DIFFERENT IF THE "SUBSTANTIAL EVIDENCE" STANDARD OF REVIEW HAD BEEN APPLIED	[93]
	A. The Standard Of Review Issue Should Not Be Decided Where Resolution Of The Issue Would Be Irrelevant To The Outcome Of The Case Being Reheard	[93]
	B. The Board's Determination Of Obviousness Is Unsupported By Substantial Evidence, As Well As Being Clearly Erroneous	[94]
III.	THE PETITION DOES NOT DEMONSTRATE THAT THIS CASE INVOVLES A QUESTION OF EXCEPTIONAL IMPORTANCE	[97]
	A. The Standard Of Review Question Does Not Impact The Balance Of Rules In the Patent System	[97]
	B. The Commissioner's Failure To Seek Application Of APA § 706 Standards Of Review For Over 45 Years Severly Undercuts His Current "Sky Is Falling" Posture	[98]

Table of Contents—Continued: Page

C. PTO Findings Of Fact Already Receive Significant Deference ..... [101]
IV. THE PETITION FAILS TO SHOW THAT THIS COURT ERRED BY APPLYING THE CLEARLY ERRONEOUS STANDARD ..... [102]
CONCLUSION AND STATEMENT OF RELIEF SOUGHT ..... [102]

TABLE OF AUTHORITIES

Cases: Page

American Paper Inst. v. American Elec. Power Serv. Corp., 461 U.S. 402 (1983) ..... [ ]
American Potash & Chem. Corp. v. United States, 402 F.2d 1000 (Ct. Cl. 1968) ..... [ ]
Anderson v. Beatrice Foods Co., 900 F.2d 388 (1st Cir. 1990) .....
Heidelberger Druckmaschienen AG v. Hantscho Commercial Prods. Inc., 21 F.3d 1068, 30 USPQ2d 1377 (Fed. Cir. 1993) ..... [ ]
In re Altenpohl, 500 F.2d 1151, 183 USPQ 38 (CCPA 1974) ..... [ ]
In re Brana, 51 F.3d 1560, 34 USPQ2d 1436 (Fed. Cir. 1995) ..... [ ]
In re Carreira, 532 F.2d 1356, 189 USPQ 461 (CCPA 1976) ..... [ ]
In re Dossel, No. 96-1340, 1997 WL 275006 (Fed. Cir. May 27, 1997) ..... [ ]
In re Kemps, 97 F.3d 1427, 40 USPQ2d 1309 (Fed. Cir. 1996) ..... [ ]
In re Lueders, 111 F.3d 1569, 42 USPQ2d 1481 (Fed. Cir. 1997) ..... [ ]
In re MacDermid, 111 F.3d 890, 42 USPQ2d 1479 (Fed. Cir. 1997) ..... [ ]
In re McCarthy, 763 F.2d 411, 226 USPQ 99 (Fed. Cir. 1985) ..... [ ]
In re Napier, 55 F.3d 610, 34 USPQ2d 1782 (Fed. Cir. 1995) ..... [ ]
In re Waymouth, 486 F.2d 1058, 179 USPQ 627 (CCPA 1973) ..... [ ]
In re Zurko, 111 F.3d 887, 42 USPQ2d 1476 (Fed. Cir. 1997) .....
Kale v. Combined Ins. Co. of Am., 924 F.2d 1161 (1st Cir.), *cert. denied*, 502 U.S. 816 (1991) ..... [ ]

Page

Singleton v. Wulff, 428 U.S. 106 (1976) ..... [ ]
Tandon Corp. v. United States Int'l Trade Comm'n, 831 F.2d 1017, 4 USPQ2d 1283 (Fed. Cir. 1987) ..... [ ]
United States v. LePatourel, 571 F.2d 405 (8th Cir. 1978) ..... [ ]
United States v. Sutherland, 428 F.2d 1152 (5th Cir. 1970) ..... [ ]

Statutes And Regulations

5 U.S.C. §§ 551 *et seq.* ..... [ ]
5 U.S.C. §706(2)(A) ..... [ ]
5 U.S.C. §706(2)(E) ..... [ ]
35 U.S.C. § 1 ..... [ ]
35 U.S.C. § 103 ..... [ ]
35 U.S.C. §135(c) ..... [ ]
35 U.S.C. §141 ..... [ ]
Fed. R. App. R. 30(a) ..... [ ]
Fed. R. App. R. 35(a) ..... [ ]
Fed. R. App. R. 40(a) ..... [ ]
Fed. R. App. R. 52(a) ..... [ ]

Page

Other Authorities

1 Jacob A. Stein et al., *Administrative Law* §1.01[4] (1997) ..... [ ]
Administrative Procedure In Government Agencies: Final Report . . ., 77th Cong., 1st Sess., S. Doc. No. 8 (1941) ..... [ ]
Administrative Procedure: Hearings on S. 674, S. 675, And S. 918 Before A Subcomm. Of The Senate Comm. On The Judiciary, 77th Cong., 1st Sess. 201 (1941) ..... [ ]
Hon. Paul R. Michel, *Appellate Advocacy— One Judge's Point Of View*, 1 Fed. Cir. Bar. J. (Summer 1991) ..... [ ]
Donald R. Dunner et al., *A Statistical Look at the Federal Circuit'sPatent Decisions: 1982-1994*, 5 Fed. Cir. B.J. 151 (Summer 1995) ..... [ ]
Attorney General's Manual On The Administrative Procedure Act (1947) ..... [ ]
P.J. Federico, *Commentary On The New Patent Act, reprinted in* 75 J. Patent And Trademark Off. Soc'y 161 (March 1993) ..... [ ]
George B. Sheperd, *Fierce Compromise: The Administrative Procedure Act Emerges From New Deal Politics*, 90 Nw. U.L. Rev. 1557 (Summer 1996) ..... [ ]
S. Rep. 2169, 1962 *U.S.C.C.A.N.* 3286 ..... [ ]
U.S. Patent and Trademark Office, *Managing Change for Global Challenges*, 102 tbl. 2 (1996) ..... [ ]
2 Who's Who In America (51st ed. 1997) ..... [ ]

INTRODUCTION

The Commission's Corrected Petition for Rehearing and Suggestion for Rehearing *In Banc* (the "Petition") is fundamentally flawed and should be denied. First, the Petition is premised on new issues that were not raised much less briefed before. Second, the Petition does not demonstrate that the outcome of this case would be different if the Court had reviewed the Board's fact findings under a "substantial evidence" standard. Third, the Petition does not establish that this case is one of "exceptional importance." Finally, the Petition does not show why the holding of this Court in *In re Lueders* should be overturned.

I. THE PETITION IMPROPERLY RAISES ISSUES *FOR THE FIRST TIME ON REHEARING*

In its original brief on appeal before this Court (the "Appellee's Brief"), the Commissioner argued that the standard of review for the fact findings of the Board of Patent Appeals and Interferences (the "Board") should be the "arbitrary and capricious" standard set forth in §706(2)(A) of the Administrative Procedure Act (the "APA"). (Appellee's Brief at 7). In the Petition, however, the Commissioner jettisons that argument and raises a new issue—that the standard of review should be the "unsupported by substantial evidence" standard set forth in §706(2)(E) of the APA. (Petition at 2-10). This attempt to raise a new issue when seeking rehearing is wholly improper and by itself warrants denial of the Petition. *See Kale v. Combined Ins. Co. of Am.*, 924 F.2d 1161, 1169 (1st Cir.), *cert. denied*, 502 US. 816 (1991); *United States v. Lewis*, 798 F.2d 1250, 1250 (9th Cir. 1986); *United States v. LePatourel*, 571 F.2d 405, 410-11 (8th Cir. 1978); *United States v. Sutherland*, 428

F.2d 1152, 1158 (5th Cir. 1970) ("Having tried and appealed its case on one theory, an unsuccessful party may not then use a petition for rehearing as a device to test a new theory.").

Having had the sole grounds of rejection under 35 U.S.C. § 103 reversed by this Court, the Commissioner also improperly attempts to raise a wholly new obviousness issue. The Petition argues for the first time that *Zurko, et al.* Claims 1, 4 and 5 are unpatentable under 35 U.S.C. § 103 over the combination of UNIX and FILER2 in view of Johri *et al.* (U.S. Patent No. 4,918,653), the 1983 IEEE article, the 1988 IEEE symposium paper and the "KSOS" article. (Petition at 13-14). The last four references were of record but were never applied in any rejection of the Claims. See Final Rejection of Examiner [A116-117]; the Board's Decision [A1-10]; Examiner's Answer [A115-122]; Examiner's Supplementary Answer [A123-127]; Appellants' and Appellee's Briefs; and this Court's opinion in *In re Zurko*, 111 F.3d 887, 42 USPQ2d 1476 (Fed. Cir. 1997). The only issue until now was whether *Zurko, et al.* Claims 1, 4 and 5 were unpatentable under 35 U.S.C. § 103 over the combination of UNIX and FILER2.

It is inappropriate, at this late stage, for the Commissioner to present an entirely new obviousness theory based on new references.

> This court has consistently refused to consider new issues on appeal because of the provisions of 35 USC 132. *See In re Altenpohl*, 500 F.2d 1151, 183 USPQ 38 (CCPA 1974). If a new rationale or reason advanced by the board requires an opportunity for applicants to respond, *In re Waymouth*, 486 F.2d 1058, 179 USPQ 627 (CCPA 1973), a new rationale or reason advanced by this court would *a*

> *fortiori* deny appellants procedural due process and amount to an impermissible new ground of rejection. Also, it would deprive appellants of the "hearing" provided by 35 USC 143 and 144.

In re Carreira, 532 F.2d 1356, 1360, 189 USPQ 461, 464 (CCPA 1976) (Miller, J., concurring). *Accord*, *Singleton v. Wulff*, 428 U.S. 106, 120 (1976) (refusing to consider new issues on appeal is "'essential in order that parties may have the opportunity to offer all the evidence they believe relevant to the issues . . . [and] in order that litigants may not be surprised on appeal by final decision there of issues upon which they have had no opportunity to introduce evidence.'" (citation omitted)). Since the Commissioner could not have asserted these new references in the appeal, he cannot assert them in the Petition:

> [A]n appellant cannot resurrect an omitted claim "merely by referring to it in a reply brief or oral argument." *A fortiori*, a party cannot be permitted to raise a new issue for the first time on a petition for rehearing in the court of appeals.

Anderson v. Beatrice Foods Co., 900 F.2d 388, 397 (1st Cir. 1990); *American Potash & Chem. Corp. v. United States*, 402 F.2d 1000, 1002 n.5 (Ct. Cl. 1968).

The proper procedure would have been for the Commissioner to seek remand to the Board for consideration of the additional references. *See, e.g., In re Dossel*, No. 96-1340, 1997 WL 275006, at *5-6 (Fed. Cir. May 27, 1997). Allowing the Commissioner to assert these new references now would deny Appellants their procedural due process rights and deprive them of a full and fair hearing on these issues. *See Carreira*, 532 F.2d at 1360, 189 USPQ at 464. Further, the Commissioner is asking

this Court to review a decision of the Board that does not exist. *See* 35 U.S.C. § 141. This is clearly unfair and inappropriate.

The Commissioner offers no reason whatsoever for not having raised earlier the "substantial evidence" standard or the additional prior art references, nor does he allege any "extraordinary circumstances" that could justify these late arguments. *See Sutherland*, 428 F.2d at 1158. Fed. R. App. P. 30(a) does not apply to the present situation as the Commissioner argues. (Petition at 13 n.6). Rule 30(a) only states that "[t]he fact that parts of the record are not included in the appendix shall not prevent the parties or the court from relying on such parts." Rule 30(a) does not permit reliance in the Petition on new references upon which the Commissioner, the Board and the examiner have never before relied.

Thus, the entire Petition is composed of arguments raised improperly for the first time on rehearing. For this reason alone, this Court should deny the Petition.

II. THE PETITION DOES NOT DEMONSTRATE THAT THIS COURT'S DECISION WOULD HAVE BEEN DIFFERENT IF THE "SUBSTANTIAL EVIDENCE" STANDARD OF REVIEW HAD *BEEN APPLIED*

A. The Standard Of Review Issue Should Not Be Decided Where Resolution Of The Issue Would Be Irrelevant To The Outcome Of The Case Be*ing Reheard*

In the Petition, the Commissioner has the burden of demonstrating that the application of the proposed new standard of review—the "unsupported by substantial evidence" standard—would result in a different outcome in this case. *In re Kemps*, 97 F.3d 1427, 1429-31,

1431 nn.5-6, 40 USPQ2d 1309, 1311-1313, 1312 nn.5-6 (Fed. Cir. 1996); *In re MacDermid*, 111 F.3d 890, 890-891, 42 USPQ2d 1479, 1479-80 (Fed. Cir. 1997). *See also In re Brana*, 51 F.3d 1560, 1569, 34 USPQ2d 1436, 1444 (Fed. Cir. 1995) (characterizing the Board's error by observing that it was "sufficiently clear that it is reversible whether viewed as clear error or as resulting in an arbitrary and capricious decision"); *In re Napier*, 55 F.3d 610, 614, 34 USPQ2d 1782, 1785 (Fed. Cir. 1995) (finding it unnecessary to address whether APA standard of review applies where court was "able to affirm the Board in this instance under the more stringent [APA] standard").

The Commissioner has failed to satisfy the burden of showing that the Board's decision would have been upheld under the substantial evidence test.

B. The Board's Determinations Of Obviousness Is Unsupported By Substantial Evidence, As Well As Being Clearly Erroneous

This is not a case where a different standard of review would have made a difference in the outcome. Had the Court applied the standard of review urged by the Commissioner, the ultimate issue would have been whether findings by the Board that the prior art teaches, either explicitly or inherently, the step of obtaining confirmation over a trusted pathway, are unsupported by substantial evidence. This issue breaks down into three subsidiary issues: (1) whether UNIX or FILER2 teaches "communicating with the user over a trusted pathway"; (2) whether UNIX, FILER2 and the record teach communicating with the user over both a trusted *and* untrusted path as claimed by Applicants; and (3) whether the problem solved by the Appellants'

invention was "previously identified anywhere in the art."

As found by the Court, "neither UNIX nor FILER2 teaches communicating with the user over a trusted pathway." *Zurko*, 111 F.3d at 889, 42 USPQ2d at 1479. The Commissioner does not deny these deficiencies in the teaching of UNIX and FILER2. Indeed, the Commissioner admits that "UNIX parses a user command outside the secure portion of the computer system (even though the command must be executed in the secure portion of the computer system). A95. It then sends the parsed command to the secure portion of the computer system for execution." *Id.* (Petition at 11) (underscoring in original).

Thus, the Board and the Commissioner admit that UNIX teaches a method where the command with the user is "outside the secure portion of the computer system." Similarly, the Board and the Commissioner's brief admit that FILER2 only teaches "a method where potentially dangerous user commands are repeated back to the user, [via an untrusted path] and a confirmation from the user is requested prior to execution." (Petition at 11-12).

Nowhere does either cited reference disclose the claimed parsing of a command in an untrusted environment; passing the parsed command to a trusted environment; repeating the command over a trusted path to the user so that the user can determine whether or not a "Trojan Horse" program changed his/her command; and then the user transmitting over a trusted path a confirmation that the repeated command represents the user's intentions. The total absence in the cited references of the claimed parsing of a command in an untrusted environment and the two-way confirmation

of the command over trusted paths precludes a finding that the references render the claimed invention obvious under 35 U.S.C. §103.

For Claim 1 to be held obvious under a "substantial evidence" standard of review, there would have to be at least some evidence in the record showing that a combination of the references can supply all elements of Claim 1. Further, the references themselves "must provide a suggestion or motivation to make such a combination." *Heidelberger Druckmaschinen AG v. Hantscho Commercial Prods. Inc.*, 21 F.3d 1068, 1072, 30 USPQ2d 1377, 1379 (Fed. Cir. 1993). Nothing in the UNIX or FILER2 references provides such a suggestion or motivation and, in any event, essential elements of the claim are entirely absent from these references.

The Commissioner has implicitly conceded that these cited references were insufficient to support a conclusion that the claimed invention would have been obvious by citing additional art, which is of record in the PTO but which was not before the Court during the initial appeal in this case. However, as discussed at 1-3 *supra*, these newly argued references may not be considered in determining whether to grant rehearing.[1]

Because there is no substantial evidence of record to support a conclusion that the claimed invention would have been obvious, the petition for rehearing should be denied, the suggestion for a rehearing *in banc* should be declined, and the case remanded to the PTO for further proceedings not inconsistent with the Court's April 15, 1997 decision in *In re Zurko*, 111 F.3d 887, 42 USPQ2d 1476.

---

[1] Even so, the additional references cited by the Commissioner suffer from the same deficiencies as UNIX and FILER2.

III. THE PETITION DOES NOT DEMONSTRATE THAT THIS CASE INVOLVES A QUESTION OF EXCEPTIONAL IMPORTANCE

Fed. R. App. P. 35(a) states that a rehearing *in banc* will ordinarily not be granted unless (1) "consideration by the full court is necessary to secure or maintain uniformity of its decisions," or (2) "the proceeding involves a question of exceptional importance." The Commissioner has not alleged any conflict or inconsistency in this Court's decisions on the issue of the standard of review for the Board's fact findings. Therefore, the Commissioner must show that this issue is "a question of exceptional importance." The Commissioner has failed to do this.

A. The Standard Of Review Question Does Not Impact The Balance Of Roles In the Patent System

The Commissioner has failed to show that the question raised by the Petition is anything more than a simple procedural issue. The standard of review is not a question of first impression, does not affect the basic jurisdiction of this Court, would not affect the value or scope of patents in general, and, in short, has no significant bearing on any important policy concern of the patent system.

The Commissioner asserts that the standard of review question is "a frequently-recurring issue" (Petition at 2) and argues that the "balance of roles between the Court and the PTO in the administration of the patent system is impacted by the standard of review . . ." (Petition at 10). The only reason for the recurrence is multiple attempts by the Commissioner to inject the standard of review question into cases pending before the Court. Furthermore, any impact on the

balance of the roles would, in fact, be minor. Every year, the PTO receives over 200,000 patent applications. U.S. Patent and Trademark Office, *Managing Change for Global Challenges*, 102 tbl. 2 (1996). Only sixty to one hundred PTO decisions on applications are appealed to this Court each year and eighty percent of those are affirmed. Donald R. Dunner et al., *A Statistical Look at the Federal Circuit's Patent Decisions: 1982-1994*, 5 Fed. Cir. B.J. 151, 158, 163, 168, 173 (Summer 1995) (hereafter "Dunner"). Of those reversed, a more deferential standard of review often would not change the result. *See, e.g., In re Napier*, 55 F.3d at 614, 34 USPQ2d at 1785, *In re Brana*, 51 F.3d at 1569, 34 USPQ2d at 1444. The mere handful of cases that potentially might be affected by a change in the standard of review, therefore, could have no meaningful impact on "the balance of roles between the Court and the PTO" (Petition at 10) or on any other aspect of the patent system. Even if the number of cases potentially affected were not small, the Commissioner has made no attempt to show that a more deferential standard of review would provide any meaningful benefit outweighing the costs associated with jettisoning the present long-established review structure.

B. The Commissioner's Failure To Seek Application Of APA §706 Standards Of Review For Over 45 Years Severely Undercuts His Current "Sky Is Falling" Posture

The APA, 5 U.S.C. §§551 *et seq.*, was signed into law by President Truman in 1946. Act of June 11, 1946, c. 423, 60 Stat. 237. The patent laws then set forth in the Revised Statutes (derived from the Patent Act of 1870) were subsequently codified at Title 35 of the United States Code. Act of July 19, 1952, c. 950, 66 Stat. 792. During the extended process of drafting the 1952

Patent Act, in which the Patent Office[2] was intimately involved, the Office never sought provision for application of APA standards of review. *See In re Lueders*, 111 F.3d 1569, 1576, 42 USPQ2d 1481, 1486 (Fed. Cir. 1997) (Rich, J., co-author of 1952 Patent Act). Contemporaneous commentary by the Patent Office Examiner-in-Chief, who co-authored the 1952 Patent Act with Judge Rich, makes no mention of the APA. *See* P.J. Federico, *Commentary On The New Patent Act, reprinted in* 75 J. Pat. and Trademark Off. Soc'y 161 (March 1993) (discussing "Review of Patent Office Decisions" without mentioning or suggesting APA applicability).

More than thirty years later, the Commissioner unsuccessfully sought "rational basis" review of an obviousness determination in *In re McCarthy*, 763 F.2d 411, 412, 226 USPQ 99, 100 (Fed. Cir. 1985). This Court rejected the Commissioner's "novel" position, explaining that application of a "rational basis" standard would be "inimical to our duty to ensure the legal correctness of the Board's decisions that are appealed, a duty entrusted to the courts." *Id.* The Commissioner chose not to seek rehearing *in banc* and did not petition for writ of *certiorari* from *McCarthy*.

In a Federal Circuit appeal filed in 1993[3], the Commissioner appears to have argued for the first time that the APA's §706(2) standards of review should be applied to its patentability findings of fact. *See In re*

[2] The patent statutes were amended in 1975 to reflect the renaming of the "Patent Office" as the "Patent and Trademark Office" ("PTO"). 35 U.S.C. § 1 ("Historical and Statutory Notes").

[3] A new Commissioner of Patents and Trademarks, Bruce A. Lehman, was appointed in 1993. *2 Who's Who In America* (51st ed. 1997) at 2515.

Napier, 55 F.3d 610, 34 USPQ2d 1782. Thus, the Commissioner could have raised the issue of APA §706(2) applicability as early as 1946, but chose not to do so until 1993. Despite bearing the burden of persuading this Court to disregard *stare decisis* principles and change its long-standing practice, the Commissioner provides no justification or explanation for an over-45-year period of acquiescence with the non-APA-based "clearly erroneous" standard of review applied to PTO findings by both the Court of Customs and Patent Appeals (CCPA) and the Federal Circuit.

The Commissioner admits that between sixty and one hundred *ex parte* appeals are taken annually to the Federal Circuit from PTO decisions. (Petition at 2). The number of appeals taken to the CCPA may have been even greater.[4] Thus, the opportunities for raising the APA standard of review issue at an earlier date were legion. Yet in literally thousands of opportunities since the APA's passage in 1946, the Commissioner saw fit not to raise the issue until 1993. To now suddenly denominate an issue potentially ripe in a multitude of appeals but not raised for over 45 years as one of "exceptional importance" calls the Commissioner's characterization into question.

[4] *See* Testimony of Conder C. Henry, Assistant Commissioner of Patents, *Administrative Procedure: Hearings on S. 674, S. 675, and S. 918 Before A Subcomm. Of The Senate Comm. On The Judiciary*, 77th Cong., 1st Sess. 201 (1941) at 618 (testifying that 198 appeals from decisions of the Patent Office were taken to the CCPA under Section 4911 of the Revised Statutes during the fiscal year ending June 30, 1940).

C. PTO Findings Of Fact Already Receive Significant Deference

Findings of fact made by the PTO in its patentability adjudications are currently reviewed under the long-standing "clearly erroneous" standard of review, the same as that applied to findings of a trial court. *In re Kemps*, 97 F.3d at 1430, 1431 n.5, 40 USPQ2d at 1312, 1313 n.5; Fed. R. Civ. P. 52(a). Although admittedly somewhat less deferential than the "substantial evidence" standard, *see Tandon Corp. v. United States Int'l Trade Comm'n*, 831 F.2d 1017, 1019, 4 USPQ2d 1283, 1284-85 (Fed. Cir. 1987), the present "clearly erroneous" standard of review is anything but unrestrained:

> Typically, however, the evidence en masse was properly admitted, probative of the central issues but in conflict. In such cases, the reviewing panel is unlikely to see "clear error" in findings or rulings unless: (1) the board or trial judge is shown to have *grossly* misanalyzed the evidence; (2) the finding or ruling is illogical or implausible; or (3) as a matter of law, the facts found are simply insufficient to support the ruling. We certainly will not reverse based merely on attorney assertion that the finding or ruling was "clearly erroneous." We need a reasoned and persuasive analysis of why.

Hon. Paul R. Michel, *Appellate Advocacy—One Judge's Point Of View*, 1 Fed. Cir. Bar. J. 1, 3 (Summer 1991) (emphasis in original). Indeed, the PTO has fared remarkably well under the current "clearly erroneous" regime—approximately 80% of its patentability determinations are affirmed on appeal by the Federal Circuit. Dunner at 154.

This is clearly not "a question of exceptional importance" and, therefore, the Petition should be denied.

IV. THE PETITION FAILS TO SHOW THAT THIS COURT ERRED BY APPLYING THE CLEARLY ERRONEOUS STANDARD

Even if the standard of review question could be considered one of "exceptional importance" under Fed. R. App. P. 35(a), the Commissioner has nevertheless failed to raise any substantial question as to the validity of this Court's long-standing rule that the "clearly erroneous" standard should apply to review of the Board's findings of fact. *See* Fed. R. App. P. 40(a). The Commissioner's arguments in favor of adopting the APA's "substantial evidence" standard and challenges to this Court's forceful reasoning in *In re Lueders* are not persuasive.

The Commissioner cites *American Paper Inst. v. American Elec. Power Serv. Corp.*, 461 U.S. 402 (1983), which involved a challenge to the Federal Energy Regulatory Commission's ("FERC") exercise of its rule-making authority, for the proposition that "when a statute that provides for review of agency action does not set forth a standard of review, the judicial review provisions of the APA apply." (Petition at 2). The Commission's attempt to bootstrap a mandatory application of APA standards of review to PTO findings from the cited portions of *American Paper* is a mischaracterization of the case's holding without reference to its facts. The *American Paper* Court merely required that the Court of Appeals review FERC's exercise of its delegated authority in promulgating the "full-avoided-cost" regulation at issue under the more lenient "arbitrary and capricious" standard of 5 U.S.C. §706(2)(A) rather than the "unsupported by substantial evidence"

standard required for formal, on the record proceedings under 5 U.S.C. §706(2)(E). *American Paper*, 461 U.S. at 412-13, 413 n.7. *American Paper* did not even approach the much more fundamental threshold question raised here, *i.e.*, whether application of *any* of the APA's §706(2) standards of review to findings of fact (in patentability adjudication, not rulemaking) is appropriate in the first instance. *See Kemps*, 97 F.3d at 1431 n.6, 40 USPQ2d at 1312 n.6 (criticizing PTO for assuming away the "key point" whether APA "should even apply to the PTO in this case").

The legislative history culminating in passage of the APA answers that question in the negative, and powerfully contradicts the PTO's literal interpretation. Historical scholarship indicates that Congress passed the APA in order to gain control over burgeoning New Deal agencies, but intended that successfully-operated, pre-existing agencies such as the Patent Office (established in 1836) be exempted. *See generally* George B. Sheperd, *Fierce Compromise: The Administrative Procedure Act Emerges From New Deal Politics*, 90 Nw. U.L. Rev. 1557 (Summer 1996) (hereafter "Sheperd"). Proposals for administrative reform first received significant Congressional interest in 1937 after the Supreme Court refused to strike down certain New Deal proposals. *Id.* at 1561. In early 1939, conservatives introduced the highly restrictive Walter-Logan bill. *Id.* at 1593-94. The Walter-Logan included a "substantial evidence" standard of review provision. *Id.* at 1602. Notably, the Patent Office (and numerous other pre-New Deal agencies) were expressly exempted from Walter-Logan. *Id.* at 1618. The Walter-Logan bill was passed by Congress but vetoed by President Roosevelt in 1940. *Id.* at 1625.

Within a few weeks of the Walter-Logan veto, the Attorney General's Committee on Administrative Procedure submitted its nearly 500-page report to Congress. *See Administrative Procedure In Government Agencies: Final Report . . .*, 77th Cong., 1st Sess., S. Doc. No. 8 (1941). The Attorney General's report ultimately resulted in enactment of the APA. 1 Jacob A. Stein et al., *Administrative Law* §1.01[4] (1997) at 1-32 (hereafter "Stein"). The report included three bills subsequently introduced into Congress. One of these, the "minority bill", S. 674, specified, *inter alia*, that reviewing courts would affirm only those agency decisions that were supported "upon the whole record, by substantial evidence." Sheperd at 1636. All three bills were extensively debated at Senate hearings, but due to wartime matters Congress took no action until 1946 when the bills were reintroduced as the compromise McCarran-Sumners Bill. 1 Stein at 1-35. The McCarran-Sumners bill with further modifications was subsequently enacted as the Administrative Procedure Act of 1946. *Id.*

Notably, the Patent Office was the sole federal agency to request exemption from the bills recommended in the Attorney General's 1941 report. Sheperd at 1638. Testifying on May 6, 1941, Conder C. Henry, Assistant Commissioner of Patents, told the Senate Judiciary subcommittee members that "[w]e accordingly strongly recommend that any bill reported out of this committee exempt by express terms the Patent Office from its main provisions." *Administrative Procedure: Hearings on S. 674, S. 675 And S. 918 Before A Subcomm. Of The Senate Comm. On The Judiciary, supra*, at 620. Henry further testified that the Patent Office had never been consulted in the preparation of the bills:

> All of these bills prima facie carry the presumption that the Attorney General's Committee made a careful investigation of the practices and procedures of each and every bureau and agency of the Federal Government in which adjudicative decisions are rendered, and that only after such investigation were these bills drafted.
>
> No such presumption should attach to the bills insofar as the Patent Office is concerned. Neither the committee nor any of its members consulted or advised with the Patent Office with respect to either of these bills [S. 674 and S. 918], and I am informed that there was no intention to do so. A[s] I have previously intimated, the majority of the committee certainly had the statutory procedure of the Patent Office in mind because they expressly excluded the Patent Office from the main provisions of S. 675. By doing so, they apparently intended those sections of the Revised Statutes governing procedures in the Patent Office, plus the provisions of the majority bill [S. 675], to govern all administrative adjudications.

Id. at 619.[5]

Post-APA enactment sources are consistent with the pre-enactment legislative history. The Department of Justice's 1947 analysis of the APA demonstrates that Congress did not intend the APA to govern review of PTO patentability determinations. *See Attorney General's Manual On The Administrative Procedure Act* (1947) at 101, n.15 ("In the Attorney General's memo-

[5] The Commissioner now correctly admits that the Patent Office was "not studied when drafting the APA." (Petition at 4 n.2).

randum to the Senate Committee, he stated that "'Courts'" includes the Tax Court, Court of Customs and Patent Appeals, the Court of Claims, and similar courts. This act does not apply to their procedure nor affect the requirement of resort thereto.' Sen. Rep. p. 38 (Sen. Doc. p. 224)."), *reprinted in* 1 Stein at App. 1C-100.

The Commissioner's literal reading is further contradicted by Congress's repeated revision of the patent laws since 1952 without ever inserting a global requirement for APA review. *See Lueders*, 111 F.3d at 1576-77, 42 USPQ2d at 1486-87 (reviewing 1962 and 1980 revisions to 1952 Act). Congress revised 35 U.S.C. §141 (Federal Circuit review of Board decisions) twice without referring to the APA: in 1982 (substituting Federal Circuit for CCPA) and 1984 (substituting Board of Patent Appeals and Interferences for Board of Patent Appeals). Other portions of the Patent Act have been revised in substantively significant fashion, *e.g.*, in 1994 (GATT/TRIPS implementation) and 1995 (Biotechnological Process Patents Act). At none of these junctures, however, was APA review of PTO findings ever incorporated into the statute, and no such legislation has yet been introduced.

Congress's addition to 1962 of the *sole* reference in Title 35 to the APA, *see* 35 U.S.C. § 135(c) (providing in connection with requirement for filing interference settlements with the PTO that "any discretionary action of the Commissioner under this subsection shall be reviewable under section 10 of the Administrative Procedure Act") strongly supports the inference that Congress does *not* intend the APA to apply to other provisions of the Patent Act such as 35 U.S.C. §141 (Federal Circuit review of PTO Board decisions). *See* 2a

Sutherland Statutory Construction §§ 46.05 (5th ed. 1992) (requiring under the doctrine of "whole statute" interpretation that "each part or section should be construed in connection with every other part or section so as to produce a harmonious whole."), 47.23 (providing pursuant to the doctrine of *expressio unius est exclusio alterius* that where a form of conduct is specified, an inference exists that all omissions should be understood as exclusions).

The "discretionary action" mentioned in 35 U.S.C. §135(c) most likely refers to exercise of the Commissioner's authority to grant leave to file a settlement agreement out of time for good cause shown. *See* S. Rep. 2169, 1962 *U.S.C.C.A.N.* 3286, 3289 (letter of Acting Deputy Attorney General Katzenbach suggesting Commissioner be given such authority "[i]n order to protect innocent parties who inadvertently fail to file a copy of their settlement agreement . . ."). That Congress designated such ministerial, fairness-based decisions as reviewable under the APA in no way suggests or implies that Congress intended similarly heightened deference for PTO patentability findings.

The Commissioner, therefore, has failed to present any significant challenge to the legal and policy foundations supporting this Court's long-standing application of the "clearly erroneous" standard of fact findings of the Board. There is no reason, therefore, for the Court to hear this issue *in banc* and the Petition should be denied.

CONCLUSION AND STATEMENT OF RELIEF SOUGHT

For the reasons set forth above, Appellants respectfully request that the Court deny the Commissioner's petition for rehearing and suggestion for rehearing *in*

banc and remand this case to the PTO for disposition not inconsistent with the Court's April 15, 1997 decision in *In Re Zurko*, 111 F.3d 887, 42 USPQ2d 1476.

Respectfully submitted,

/s/ A. SIDNEY JOHNSTON

Date: June 20, 1997

A. SIDNEY JOHNSTON
Corporate Law Department
Digital Equipment
Corporation
111 Powdermill Road
Maynard, Massachusetts
01754-1499
(508) 493-6526

Attorney Of Record For
Appellants

Of Counsel:

Ronald C. Hudgens
Digital Equipment Corporation
(508) 493-5493

Janice M. Mueller
Assistant Professor Of Law
Suffolk University Law School
(617) 573-8171

John F. Sweeney
Israel Blum
Michael O. Cummings
Morgan & Finnegan, L.L.P.
(212) 758-4800